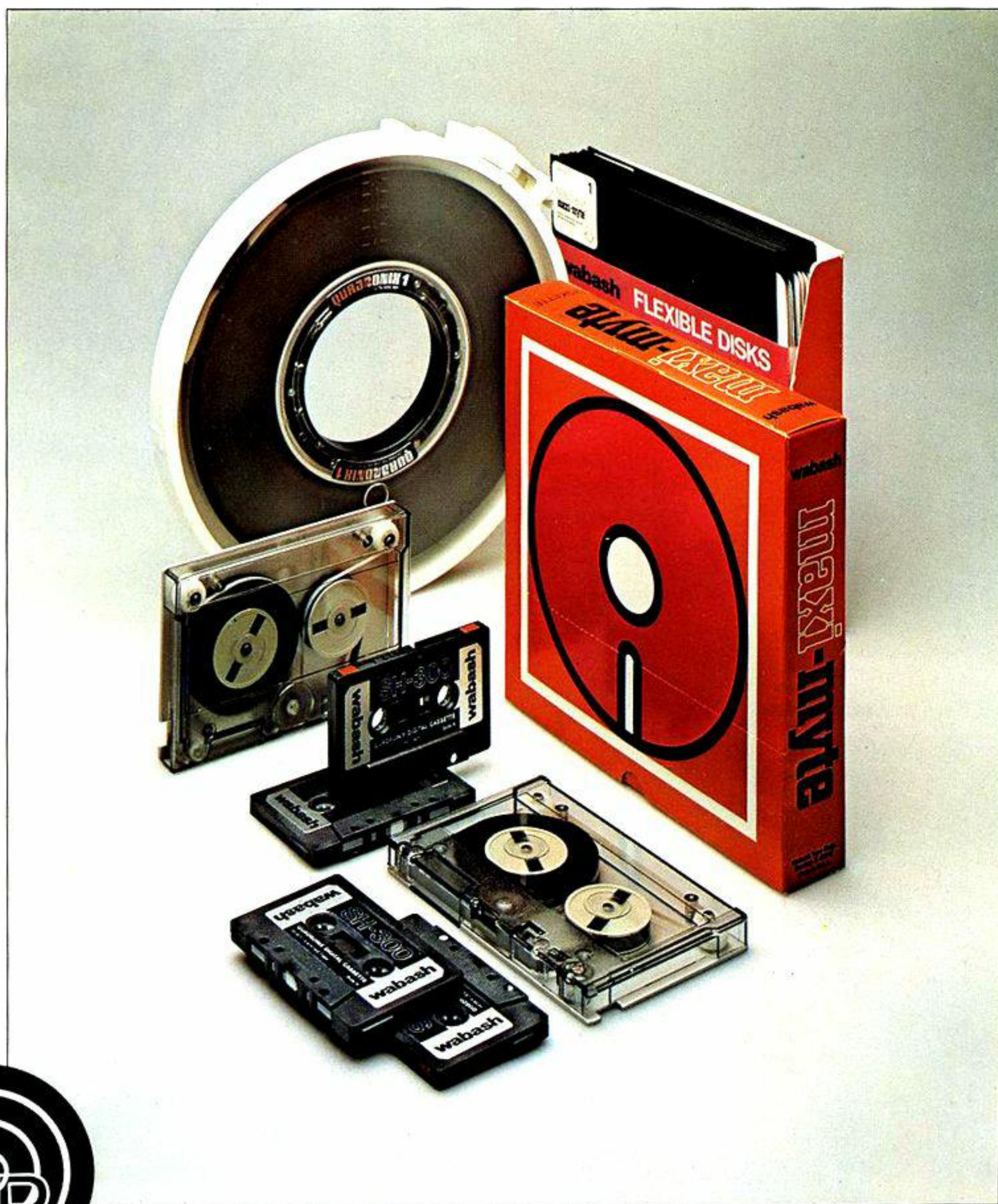
Sai chi è il distributore per l'Italia dei prodotti magnetici Wabash?

A.P. di Adduci e Ponza

20137 MILANO - VIA MONTE LEPINI, 4 - TELEFONO (O2) 54.84.097 - 54.84.171

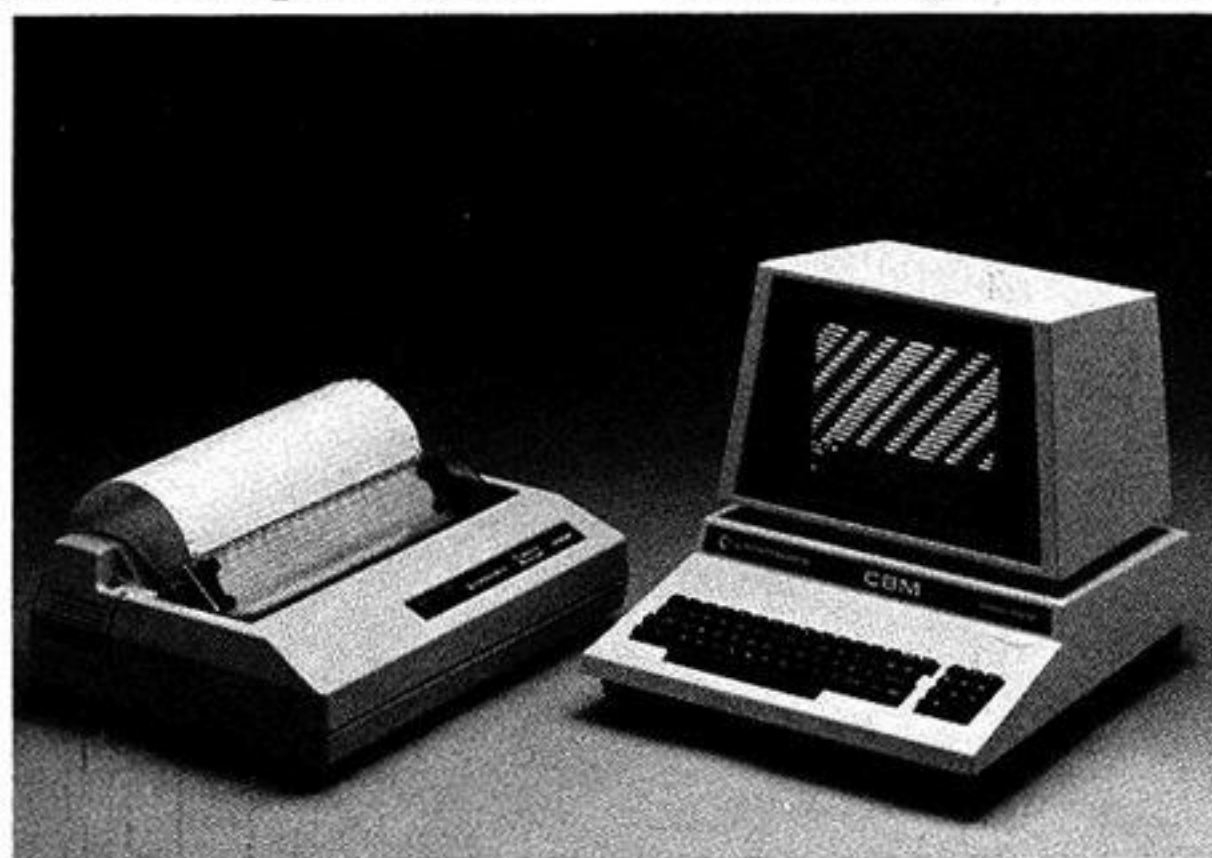
DISTRIBUTORI REGIONALI - Lazio: APC/Roma, Tel. (06) 8392646 - Piemonte-Aosta: YORK DATA/Torino, Tel. (011) 489764 - Sicilia: E.d.m./Palermo, Tel. (091) 580419 - Toscana: C.S.S./Firenze, Tel. (055) 679630 - Veneto: F.C.M./Altavilla -Vi- Tel. (0444) 566225 - Puglie: Librocart/Putigliano Tel. (080) 734037

SOMMARIO

Direttore responsabile: Michele Di Pisa **Redazione:** Alessandro De Simone, Franco Rao.
Direzione, redazione: Piazza Arduino, 3 - 20149 Milano - tel. (02) 434354 - 435717 -
Pubblicità: Milano: Paola Bevilacqua, Françoise De Massiac, Nadia Peroni, Tina Ronchetti, Villa Claudio - Piazza Arduino, 3 - tel. (02) 434354 - 435717 - Torino: Spazioblù di Daniela M. Costamagna - Via Filadelfia, 50 - 10134 Torino - tel. (011) 327617 - Roma: Rosanna De Marinis - Via Piero Foscari, 70 - Roma - tel. (06) 8109679
Composizione: Minisystems Italia - **Selezioni:** Org. Aldo Ghiacci -
Stampa: CoPeCo - Via Figino, 20 - Pero - Suppl. a Computer - Registrazione Tribunale di Milano n. 993 del 3/12/79 - Spedizione in abb. post. grup. III n. 70 - Pubblicità inferiore al 70 %

Commodore è alla Homic

Vieni alla Homic, e fatti mostrare un "personal" Commodore: li trovi tutti, dall'eccezionale Vic20 Colour Computer, che permette di lavorare con 24 colori, produce suoni e musica ed è collegabile con ogni apparecchio televisivo e risolve problemi scolastici, di divertimento e tecnico scientifici, alle Serie CBM destinate a trattare quantità medie e grandi di dati per la gestione della casa, degli studi professionali e delle piccole aziende.

Vieni alla Homic: trovi il meglio.

HOMIC

Uffici: Piazza de Angeli, 3 - Milano - Tel. 4695467-4696040-4984583

MP

Sono interessato a ricevere materiale illustrativo

nome ____________________

indirizzo ____________________

Invito al club

Il Pet è stato il primo personal sbarcato nel nostro Paese. In pochi anni i prodotti della Commodore Business Machines si sono diffusi in tutta la Penisola, facendo scoprire i vantaggi ed il gesto del personal computing a migliaia di italiani. Oggi il 30% circa dei possessori di personal computer hanno un prodotto CBM, dal piccolo VIC ai più potenti 8032.

Era naturale, dunque, che fosse dedicata a questa famiglia di macchine la prima rivista italiana di personal computing mono-marca.

All'estero (ed il pensiero corre subito agli Usa) le riviste di informatica personale dedicate esclusivamente a questa o a quella famiglia di microcomputer pullulano. Alcune competono per diffusione e volume anche con le più prestigiose pubblicazioni a carattere generale. In realtà le completano, dando al lettore utente d'una specifica macchina, informazioni e programmi che difficilmente riuscirebbero a trovare spazio in una rivista di tipo "universale".

Trasferendo al settore editoriale una tecnologia tipica dell'edp, si può dire che questo tipo di pubblicazione rappresenta la terza generazione delle riviste di informatica. La prima è costituita dalle testate "general-purpose" come *Datamation* e *ComputerWorld* negli Usa, o *Computer, Liner Edp, Informatica Oggi* e *ZeroUno* in Italia. La seconda generazione è rappresentata dalle pubblicazioni di personal computing in senso generale (ricordiamo le americane *Byte, Microcomputing, Interface Age,* ecc. e le italiane *Micro & Personal Computer, Bit, MC Microcomputer).*

All'estero gli esempi di riviste di informatica della terza generazione sono numerosissimi. Per rimanere nell'ambito delle pubblicazioni dedicate ai sistemi Commodore ricorderemo soltanto le americane *Power & Supply* e *Commodore Compute,* l'inglese *VIC Computing,* la tedesca *Computer Journal.*

Commodore Computer Club si inserisce in questo filone.

Naturalmente Commodore Computer Club non vuole essere soltanto una rivista *per* gli utenti di sistemi CBM. La sua ambizione principale è diventare rapidamente la pubblicazione *degli* utilizzatori di queste macchine stupende, un punto di incontro per mettere a confronto le varie esperienze, attingere idee e suggerimenti, una fonte continua di listati, e routine di programmazione a basso costo.

La collaborazione dei lettori è perciò basilare per il successo dell'iniziativa, così come è importante la collaborazione di quanti operano professionalmente con i sistemi Commodore. Da essi attendiamo notizie, informazioni, articoli, valutazioni sui programmi-prodotto provati.

La rivista, come dice il suo nome, vuole essere un club. E come in ogni club tutti i frequentatori debbono contribuire al suo successo.

Michele Di Pisa

L'autore di "Aria", "Non mi lasciare" e "Tu cosa fai stasera" oltre che per la gestione dell'archivio nastri ed il calcolo della SIAE, utilizza il computer per annotare tutte le sue idee musicali. Il ruolo del CBM 3016 nella preparazione di "Spirito della terra".

Dario Baldan Bembo compu-cantautore

Chi sono gli utilizzatori di personal computer? E in particolare, chi compra ed usa le migliaia di sistemi VIC e PET che si vendono con ritmo sempre crescente nel nostro Paese?

Studenti, si dice, o piccole aziende. In parte è vero. Ma accanto ad essi incontri personaggi particolari, uomini e donne del mondo dello spettacolo, dello sport, della cultura, scrittori famosi e generali che contano, medici, avvocati e architetti di prestigio: persone, insomma, che pur vivendo negli ambienti più diversi sono accomunati dal fatto di appartenere allo stesso club ideale degli appassionati di personal computing.

"ARIA" è il titolo dell'LP che ha portato alla ribalta del mondo della musica Dario Baldan Bembo. Da allora sono trascorsi sette anni e nelle sue 110 versioni questo disco, in tutto il mondo, è stato venduto in oltre 10 milioni di esemplari.

Oggi, Dario ha 34 anni, ha inciso altri tre LP (quello che ha riscosso maggiore successo si intitola "Non mi lasciare"), s'è piazzato terzo al festival di San Remo 1981 con la bellissima "Tu cosa fai stasera" ed ora si prepara al lancio della sua ultima incisione, "Spirito della terra", un disco per il quale ha avuto bisogno, oltre che della sua indiscussa fantasia musicale, di tutto l'aiuto di un CBM 3016 e d'una scenografia ed ambientazione naturale che solo il Padreterno poteva approntare: un prato in mezzo a un bosco vicino Maggiora in provincia di Novara.

"Spirito della terra"

La fama di Dario Baldan Bembo cantautore spesso è offuscata da quella di Dario Baldan Bembo compositore. Gran parte della sua produzione musicale, infatti, viene "messa in scena" da altri cantanti, come Renato Zero, nel cui "file" canoro figurano almeno tre titoli di Dario: Amico, Più sù e Stranieri.

Per te riservi solo alcune delle tue composizioni. Perchè?

"Mi piace preparare bene le cose," risponde, "piuttosto che inflazionare". Così la preparazione di "Spirito della terra" è stata lunga e meticolosa. Soprattutto a causa dell'originalità tecnico-musicale del disco, che non è stato registrato, com'è regola generale, in sala di incisione, ma all'aperto, in uno squarcio di prato in mezzo al bosco di Maggiora.

La ragione?

"Volevo suoni purissimi, senza riflessioni ambientali di pareti, utilizzando echi naturali invece di quelli riprodotti artificialmente. Quest'idea l'avevo in mente da almeno due anni, ma non è stato facile convincere la casa discografica. Registrare fuori d'una sala d'incisione significa dover trasportare tonnellate e tonnellate di strumenti musicali e d'apparecchiature sofisticate, creare un mini-villaggio turistico per gli orchestrali e i tecnici, essere certi che quella purezza di suono che ci riproponevamo non venisse turbatada fattori imprevisti.

Per convincere i dirigenti della mia società ad accettare la proposta, il mio 3016 mi è stato di enorme utilità. Con esso, infatti, ho potuto elaborare e confrontare due budget di spesa alternativa: uno per l'effettuazione della registrazione in sala incisione (prevedendo, quindi, il noleggio della sala); l'altro per l'ipotesi aria libera, sostituendo l'onere della sala con quelli per i trasporti ed il soggiorno delle persone. I due totali erano molto simili. Ma i numeri non potranno mai contabilizzare il vantaggio, poi sperimentato da tutti, di lavorare con gioia ed entusiasmo. Per il successo di "Spirito della terra" lo spirito con cui abbiamo lavorato sarà determinante".

Anche il CBM 3016, ha contribuito alla realizzazione di questo LP. Il calcolo comparativo delle due previsioni di spesa è un esempio. Ma, già nella fase di gestazione del lavoro, Dario Baldan Bembo si è messo parecchie volte alla tastiera del computer per...memorizzare idee e spunti musicali, utilizzando il programma che egli stesso aveva preparato

La scoperta del computer

Un cantante che smanetta sul computer, ovviamente non può essere un personaggio a senso unico. Si scopre, così, che Dario Baldan Bembo di hobby ne ha più d'uno.

"Naturalmente," dice, "amo fare del cinema e ne ho realizzato qualcuno a soggetto. Il cinema mi piace moltissimo soprattutto per il modo di coniugare musica ed immagine. Un altro hobby che mi fa infollire è il modellismo, che coltivo dall'età di otto anni. Ora posseggo 15 locomotrici e 80-100 vagoni in scala HO".

E l'interesse per il computer com'è nato?

"Per caso, entrando in un negozio. Avevo il problema di archiviare i nastri del mio studio, che gestivo (si fa per dire) con una miriade di foglietti volanti. Quando in uno scaffale hai 100 nastri,per trovarne uno a volte perdi ore ed ore: devi stare a scoltarli, risentirne vari brani in modo random, tornare ad ascoltarli ancora.

Non so per quale arcano ragionamento, mi ero convinto che un personal computer non potesse costare meno di 13 milioni. Quando ho scoperto che il suo prezzo era quasi un decimo non ho avuto dubbi".

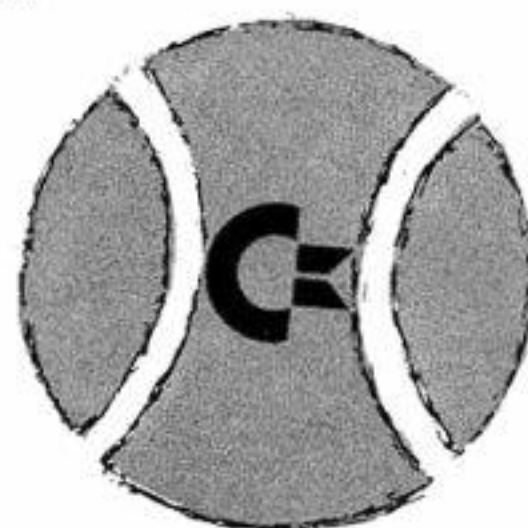
Ed hai risolto il problema dei nastri?

"Il discorso archiviazione nastri è stato l'inizio. Successivamente ho pensato di utilizzare il computer per comporre della musica ed ho fatto un programma apposito. Ma le melodie che ne ottengo sono semplicemente impossibili. Io sono un cantante melodico: guai se utilizzassi le musiche ottenute col mio programma di composizione casuale".

Il primo personal computer che hai visto non era un Commodore. Perchè, allora, hai scelto il 3016?

"Perchè dal punto di vista

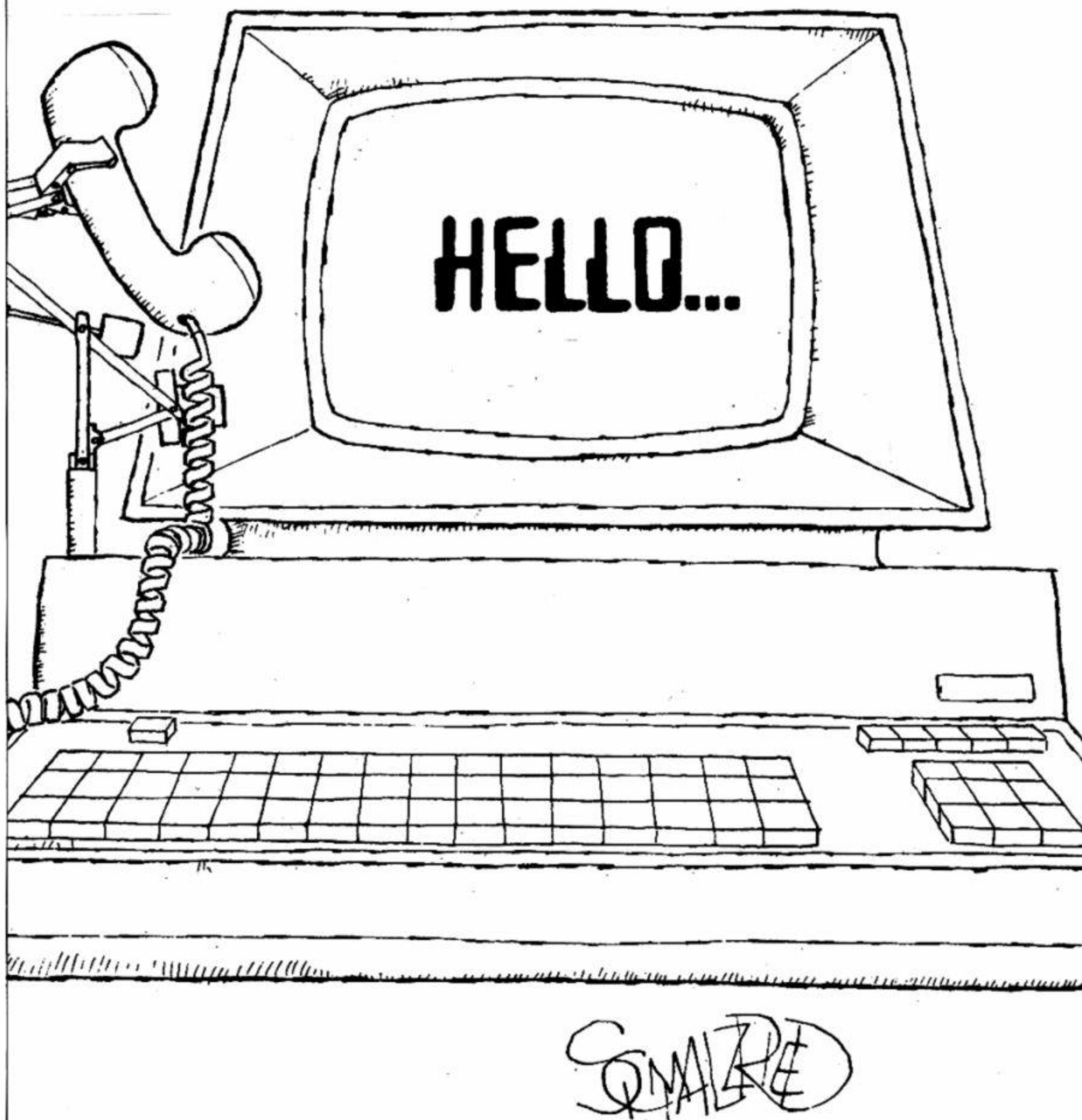
“Mega power” Commodore.
per espandere il tuo CBM

...a così.

Da così...

Se hai un CBM 4000 oppure 8000 approfitta del “Mega power” Commodore.
Un modo esclusivo per megaespandere la capacità dei tuoi sistemi Commodore.
Per esempio col DUAL FLOPPY 8250, a meno di 3.500.000* aggiungi 2 Mega-Bytes. Vuoi di più?
Con l'HARD DISK 9060 aggiungi 5 mega-bytes, con meno di 5.000.000*.
E con 6.100.000* hai sotto controllo altri 7,5 Mega-bytes: basta scegliere l'HARD DISK 9090.
Certo, per essere avanti agli altri bisogna dare cose avanti...

Per avere l'indirizzo del rivenditore Commodore rivolgiti al tuo Distributore Commodore regionale:

LIGURIA - PIRISI INFORMATICA - Piazza Cavour, 19 - 16043 Chiavari - Tel. 0185/30.10.31 • PIEMONTE - ABA ELETTRONICA DI CARAMIA - Via Fossati, 5/C - 10141 Torino - Tel. 011/33.20.65 • LOMBARDIA, TRENTINO - HOMIC PERSONAL COMPUTER srl - Piazza de Angeli, 3 - 20146 Milano - Tel. 02/498.82.01 • VENETO, FRIULI, VENEZIA-GIULIA - CO.R.EL. Friuli Computers - Via Mercatovecchio, 28 - 33100 Udine - Tel. 0432/29.14.66 • EMILIA-ROMAGNA, MARCHE - S.H.R. srl - Via Faentina, 175/A - 48010 Fornace Zarattini (Ravenna) - Tel. 0544/46.32.00 • TOSCANA - M.C.S. SpA - Via Pier Capponi, 87 - 50132 Firenze - Tel. 055/57.13.80 • LAZIO, UMBRIA - KIBER ITALIA srl - P.le Asia, 21 - 00144 Roma EUR - Tel. 06/59.16.438 - ATLAS SYSTEM srl - Via Guglielmo Marconi, 17 - 01100 Viterbo - Tel. 0761/22.46.88 • ABRUZZO - PRAGMA SYSTEM srl - Via Tiburtina, 57 - 65100 Pescara - Tel. 085/50.883 • CAMPANIA - GRAAL SYSTEMS Elaboratori Gestionali - Via P. Grisignano, 4 - 84100 Salerno - Tel. 089/32.17.81 • PUGLIA - MASELLI X L'UFFICIO - Via L. Zuppetta, 5 - 71100 Foggia - Tel. 0881/76.1.11 - BUSINESS AUTOMATION SYSTEM srl - Largo De Gemmis, 46/B-46/C-48-48/A-48/B - 70124 Bari - Tel. 080/22.75.75-22.73.44 • CALABRIA - SIRANGELO COMPUTERS srl - Via Nicola Parisio, 25 - 87100 Cosenza - Tel. 0984/75.7.41 • SICILIA - EDILCOMPUTER PROGETTI dell'Ing. Giuseppe Carbone - Via La Farina, 141 Is. I. - 98100 Messina - Tel. 090/29.28.269 • SARDEGNA - S.I.I. Sistemi Integrati Informatica - Via S. Lucifero, 95 - 09100 Cagliari - Tel. 070/66.37.46

estetico è il più bello. Le mie cognizioni tecniche, con tutta franchezza, non mi consentivano di fare una scelta ragionata. Ad esempio le capacità sonore di questa macchina le ho scoperte solo parecchi mesi dopo che ci smanettavo. Ed è stata una bella sorpresa". ▪

Alea jacta est

QUANTE volte giocando o divertendoci con passatempi che utilizzano i dadi quale determinante di avanzamento del gioco è accaduto che quei cubetti, dopo il normale lancio, cadano dal tavolo o si fermino in posizioni dubbie, spostano le pedine o che altro s c o c c i ante avvenimento? Certamente un "fracco" di volte.

Perchè allora non utilizzare questo programmino, forse un po' "scemo", ma senza dubbio valido allo scopo?

Il programma è stato studiato in modo tale che possa girare senza alcun problema sia su schermo a 40 che ad 80 colonne, e perchè no anche per il VIC, basta infatti modificare il valore della variabile CBM fornendo adeguatamente il valore necessario.

All'inizio, subito dopo il normale RUN, viene richiesto il numero totale dei dadi che devono essere utilizzati. Se volete giocare a Monopoli si dovrà battere il tasto numerico 2. In seguito il programma proseguirà sempre "lanciando" due dadi.

Per "lanciare" i dadi si dovrà premere la barra di spazio ed ecco una nuova combinazione casuale (RND), visualizzazione grafica dei dadi, la somma dei valori delle singole facce.

Chi poi pensa che il PET è un dado un po' troppo grosso per essere appoggiato su un tavolo da gioco ebbene con un cavo bifilare, un interruttore a pulsante, l'uso della User Port e la modifica della riga 190 con una P... Beh, volete che vi dica proprio tutto?

```
100 GOTO160
110 ***************************************************************************
120 *                                                                         *
130 *              LANCIO CASUALE DI  N (DA 1 A 5)  DADI                      *
140 *                                                                         *
150 *K*********************************************************************H*
160 CBM=80:REM (40 PER 4032)(80 PER 8032)
170 PRINT "▒▒▒▒▒"TAB(CBM/2-8)"▒ LANCIO DADI"
180 PRINT TAB(CBM/2-19) "▒▒▒▒NUMERO DEI DADI DA LANCIARE (1-5)?"
190 GET N$:IF N$<"1" OR N$>"5"THEN 190
200 N=VAL(N$):PRINT"▒"
210 PRINT"▒▒▒▒▒▒▒▒▒"TAB(CBM/2-17)"BATTI ▒SPAZIO▒ PER LANCIARE I DADI"
220 GET I$:IF I$<>" " THEN 220
230 GOSUB 240 : GOTO 210
240 PRINT"▒":T=0:CE=INT((CBM-N*4)/2)
250 FOR C=1 TO N
260 PRINT"▒":X=C*4-4+CE
270 D=INT(RND(1)*6)+1
280 ON D GOSUB 370,350,330,360,360,360
290 ON D GOSUB 340,370,340,370,340,360
300 ON D GOSUB 370,330,350,360,360,360
310 T=T+D : NEXT
320 PRINTTAB(CBM/2-6)"▒▒▒▒▒TOTALE ="T : RETURN
330 PRINTTAB(X)"▒●  ":RETURN
340 PRINTTAB(X)"▒ ● ":RETURN
350 PRINTTAB(X)"▒  ●":RETURN
360 PRINTTAB(X)"▒● ●":RETURN
370 PRINTTAB(X)"▒   ":RETURN
```

DOMANDE RISPOSTE

Per tutti coloro che posseggono un computer Commodore, abbiamo raccolto una serie di domande sul VIC e prodotti correlati che potranno forse insegnare qualcosa di nuovo anche a coloro che lo usano da tempo. In futuro questa rubrica è destinata ad ospitare le domande dei lettori.

Quando uscirà la guida di consultazione per il programmatore?

La guida di consultazione per il programmatore (per la precisione la "Programmer's Reference Guide") è appena uscita negli Stati Uniti ed è disponibile presso i rivenditori autorizzati della Commodore.

In che modo è possibile eliminare le linee ondulate che compaiono sullo schermo TV?

Quelle linee sono causate dalle frequenze radio, un tipo di disturbo che tutti i computer tendono a generare. Per attenuarne gli effetti (eliminarlo è impossibile), si può provare a modificare l'orientamento del modulatore rispetto all'antenna TV o a sintonizzare diversamente il televisore, facendo attenzione che l'interrutore di accensione sia opportunamente collegato.

Il mio schermo tende a sbattere rapidamente ogni volta che uso il computer. Come posso eliminare questo inconveniente?

E' un inconveniente che si verifica abbastanza sovente. Se avete un VIC 20, potete usare il comando POKE in questo modo:

POKE 36864,133

che va ripetuto dopo ogni reset (compreso quello che si verifica all'accensione).

Perchè qualche volta è così difficile caricare un programma da cassetta?

Il fenomeno può essere attribuito alle radiazioni provenienti dallo schermo TV che interferiscono sul caricamento del programma su nastro quando si usa il registratore Datassette. Per eliminare l'inconveniente, è sufficiente allontanare il più possibile il registratore dallo schermo.

Un altro consiglio utile: poichè i nuovi nastri sono spesso avvolti con forza intorno all'aspo, avvolgeteli e svolgeteli un po' di volte senza effettuare caricamenti; ciò ne provocherà un allentamento e una maggiore facilità d'uso.

Il software PET/CBM può essere adattato al VIC?

In generale sì, purchè ci sia memoria sufficiente. In realtà però i programmi che si adattano meglio sono quelli in BASIC che non contengono nè PEEK nè POKE, e poichè la maggior parte dei programmi PET/CBM ne fanno uso e le dimensioni dello schermo sono molto diverse da quelle del VIC, la cosa migliore è conservare l'idea generale del programma e riscriverlo completamente usando mappe di memoria.

La mappa di memoria del VIC è riportata nel Programmer's Reference Guide del VIC 20 che può essere richiesta alla Commodore, le mappe PET/CBM sono riportate in diverse pubblicazioni fra cui il volume "PET/CBM Personal Computer Guide" pubblicato dalla Osborne–Mc-Graw Hill.

Qual'è lo scopo della cartuccia interfaccia terminale RS232?

E' quello di convertire il segnale prodotto dal VIC in un vero formato standard RS232, usato da numerosi MODEM di altre case.

Come viene agganciato il MODEM al VIC?

L'unico modo, attualmente praticabile in Italia, è quello di acquistare una cartuccia interfaccia terminale VIC RS232 e di collegarla, tramite cavo RS232/RS232 ad un accoppiatore acustico RS232.

E' previsto un incremento del software disponibile per il VIC?

Alla Commodore esiste un gruppo di progettazione software, attualmente impegnato nello sviluppo di programmi, destinati sia al mondo del tempo libero che a quello del lavoro, che, prevediamo, saranno presto disponibili.

Come usate i "joystick" con il VIC? Ne fate uso con le i programmi di giochi?

I programmi attualmente in circolazione non ne fanno uso.

E' possibile agganciare al VIC 20 un audioregistratore a cassette?

No, l'audioregistratore a cassette non è in grado di funzionare con il VIC 20. Con qualsiasi cartuccia a nastro, è necessario usare il Datassette VIC, la quale comunque usa nastri a cartuccia. Per ottenere i migliori risultati, conviene usare il formato di nastro più corto e assicurarsi che non sia al biossido di cromo, una sostanza che taglia le alte frequenze cioè quelle a cui i dati vengono memorizzati.

E' possibile collegare al VIC più periferiche contemporaneamente?

E' possibile collegare in "daisy-chain" fino a cinque unità a disco. Nel caso in cui si voglia inserire anche una stampante, è sufficiente collegarla all'ultima unità della catena.

E' possibile collegare un monitor al VIC 20?

Sì, e senza l'uso del modulatore. Il cavo di collegamento varia in base al connettore di ingresso del monitor.

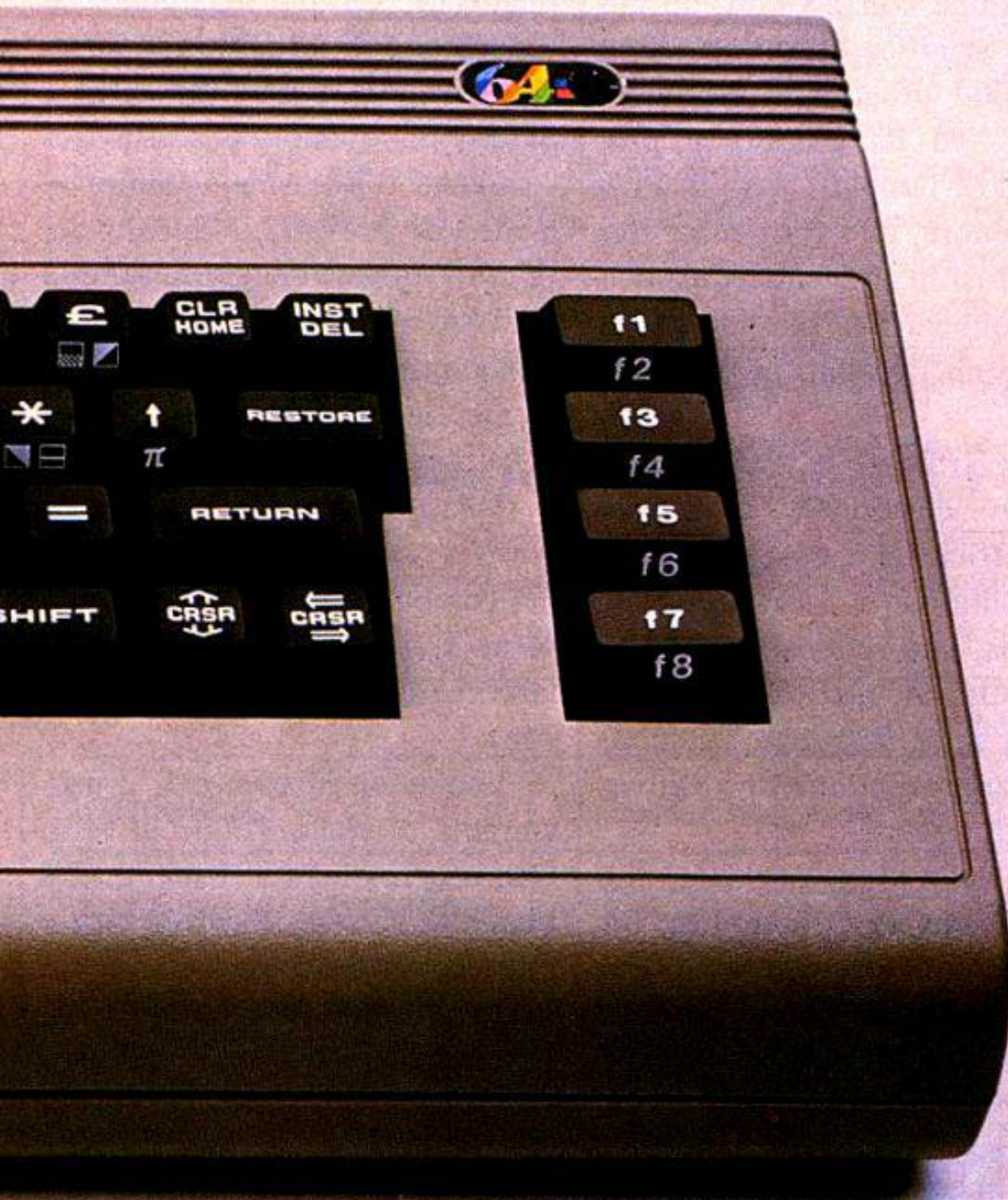
Che tipo di BASIC viene usato sul VIC 20?

Il VIC 20 usa il PET BASIC 2.0.

E' possibile aumentare il numero di colonne per linea?

Attualmente non esiste nessuna tecnica efficace. Si potrebbero scrivere degli appositi programmi, ma richiederebbero troppa memoria per risultare veramente utili.

Arriva il Commodore 64

In grado di competere con i più diffusi personal del mercato, il 64 si pone a mezza / via tra il VIC ed il Pet. Con tutti i caratteri grafici di questi modelli.

DA CHE parte cominciare a parlare di questo recente personal computer della Commodore? I suoi progettisti hanno scelto di partire dal rapporto prezzo/prestazioni. E non hanno torto, se si pensa che al prezzo di un piccolo personal, sviluppato con in mente anche il mercato dell'hobbistica e del tempo libero, ci si porta a casa una memoria incorporata di 64K, un sintetizzatore musicale a livello professionale, una grafica ad alta risoluzione in 16 colori e una tastiera con 66 tasti.

L'interesse, però, non si ferma qui. Il Commodore 64 nasce dal patrimonio di conoscenze ed esperienze che stà dietro il suo nome. Innanzitutto il software.

I programmatori della casa americana hanno esaminato la giungla di prodotti immessi sul mercato negli ultimi anni (qualcosa come 5000 programmi) e ne hanno selezionati alcuni fra i più popolari, fra cui un programma per la gestione delle mailing list, uno per l'elaborazione di testi e numerosi pacchetti di memorizzazione e richiamo di dati.

Una volta selezionato, il materiale è stato messo in mano a progettisti software indipendenti, invitati a provare la nuova macchina e ad effettuare un intervento di adattamento dei programmi scelti. Il lavoro ha dato risultati estremamente interessanti e, per la qualità raggiunta, oltre che per il numero di programmi disponibili, il software nel suo insieme può essere ritenuto fra i migliori, anche rispetto al prezzo.

Nel settore dei giochi, poi, al numero e alla complessità dei programmi, il 64 aggiunge una risoluzione grafica e una ricchezza di effetti, quali suoni e luci, fino ad oggi impensabili su un computer così piccolo.

Il Commodore 64 è inoltre compatibile con il CP/M, un nome che significa "porta aperta su altri 2000 programmi di grande utilità", ed ha accesso ai programmi sviluppati per il PET Commodore, uno fra i computer più diffusi in ambiente educativo e di conseguenza più fornito di programmi destinati agli studenti.

Per verificare la ricchezza del software disponibile, è comunque sufficiente dare un'occhiata alla Commodore Software Encyclopedia, una vera e propria enciclopedia in grado di fornire un'idea sulla lunghezza, l'ampiezza e l'approfondimento degli argomenti trattati.

Fin dalla prima occhiata, ci si accorge che il Commodore 64, non è solo un personal per il tempo libero. Tutt'altro. Nel progettarlo, si è tenuto presente il mondo aziendale e le esigenze di chi vi lavora, che vanno dall'esecuzione di calcoli all'elaborazione di testi e alla memorizzazione e richiamo dei dati. Ma tutto è stato studiato con l'obiettivo di rendere gradevole anche il lavoro più noioso. La scelta dei programmi è ricca e puntuale nel fornire la soluzione migliore così come molto efficace è la grafica nel trasformare le informazioni in istogrammi, diagrammi a torta e schemi di immediata comprensione. Ed è proprio in questo ambito che la ricca gamma di colori disponibili si rivela estremamente utile.

Nel settore del software gestionale, una particolare importanza è rivestita dagli strumenti finanziari, cioè programmi che consentono di facilitare i processi decisionali offrendo soluzioni a problemi quali: la scelta fra l'acquisto o il leasing, l'ammortamento di prestiti, il calcolo del valore attuale o dei giorni fra due date. A ciò si aggiungano poi i programmi statistici, fra cui le medie, le regressioni, il calcolo della frequenza, ecc.

Tutto questo ha però poco senso se non lo si vede inserito nel computer e non si può verificarne la facilità di impiego e di inserimento nelle abitudini di lavoro.

Vediamo, ad esempio, il programma di elaborazione di testi. Con un programma elementare, è possibile battere, memorizzare e rivedere un testo e poi stamparlo in tre formati possibili: bozza, lettera e lettera formale, secondo l'esigenza di chi legge. E se poi c'è chi non si accontenta, esiste un programma sofisticato che non ha confronti nel campo dei personal computer.

Anche per quanto riguarda la memorizzazione e il richiamo dei dati, il 64 garantisce la strada più breve per definire i dati desiderati, selezionarli e ridisporli o fare con essi qualsiasi altra cosa (modificarli, cancellarli, riprodurli in rapporti stampati, ecc.).

Consci dell'importanza che le

telecomunicazioni sono destinate a rivestire sempre di più in tutti i campi, la Commodore ha previsto la possibilità di collegare il 64 al telefono, e, attraverso il telefono, con l'immenso mondo delle informazioni del Videotel, che vanno dalle quotazioni di borsa, ai prezzi dei mercati all'ingrosso, alle prenotazioni aree e ferroviarie, ecc., e con servizi attualmente curati dagli uffici postali.

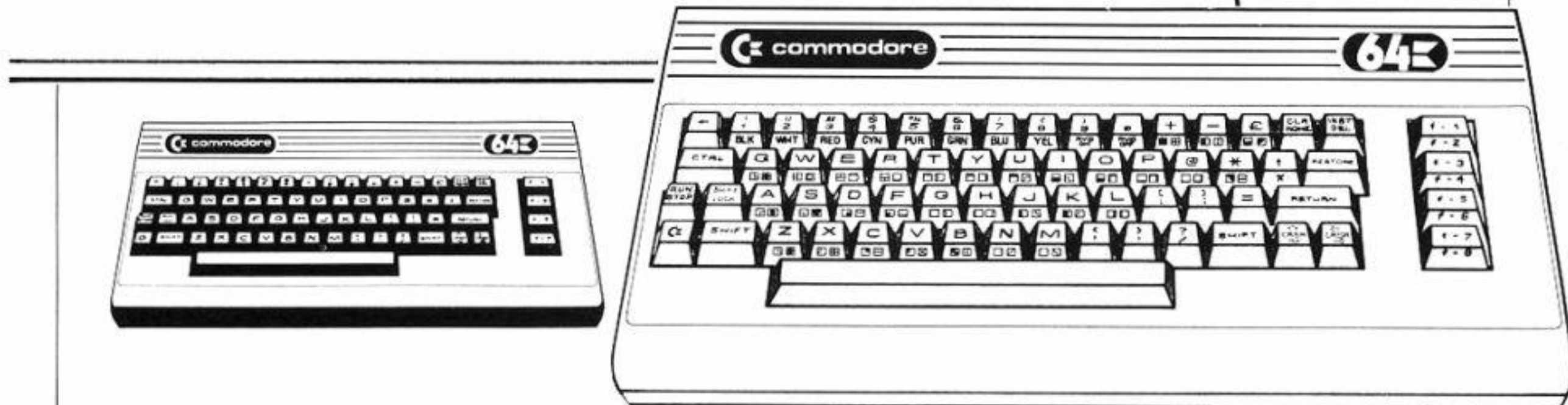
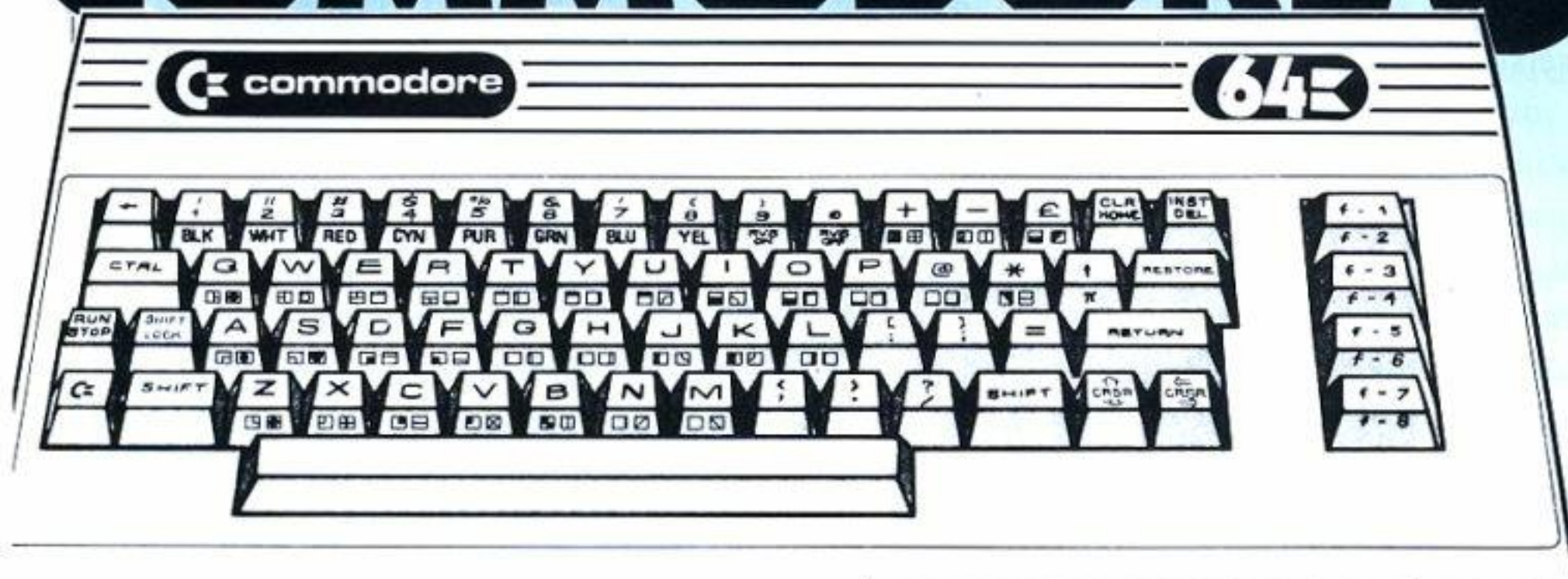
Per ora sembra fantascienza, ma le cose stanno andando sempre più in fretta e questo nuovo Commodore 64 sembra non volersi lasciare superare.

Ma veniamo alla macchina vera e propria

Particolarmente interessante è l'aspetto hardware del 64, realizzato con l'impiego di chip progettati dalla Commodore su proprie specifiche esigenze, cioè con il prodotto finito in mente.

- *Il microprocessore,* è il 6510 prodotto dalla stessa Commodore e usa il set di istruzioni del 6502 con altre linee di ingresso/uscita. Per mettere il 64 in grado di eseguire il CP/M, e avere così a disposizione tutta la libreria di programmi fino ad oggi sviluppati, è prevista un'apposita opzione con inserimento ad innesto del microprocessore Z80.
- *Quanto a memoria,* il 64 è dotato di 64K di RAM: 39K sono a disposizione dell'utente per i programmi in BASIC, 52K per i programmi in linguaggio macchina, 20K di ROM interna contengono il sistema operativo e il linguaggio BASIC Commodore.
- *La tastiera* è del tipo macchina per scrivere, con 66 tasti, con minuscole e maiuscole, e quattro tasti in funzione programmabile non assegnata. Attraverso la tastiera è possibile selezionare direttamente sia i colori che le opzioni grafiche PET disponibili.
- *I tasti* per l'editing comprendono: il tasto CTRL, per scegliere il colore dei caratteri ed eseguire i comandi speciali; il tasto RUN/-STOP, per caricare i programmi dalla cassetta; il tasto SHIFT/SHIFT LOCK, per impostare a scelta le maiuscole o le minuscole e la grafica PET; il tasto RVS ON/OFF, per invertire fra loro sul video i colori del testo e dello sfondo; il tasto CLR/HOME, per spostare il cursore nella posizione

5 Personal a confronto

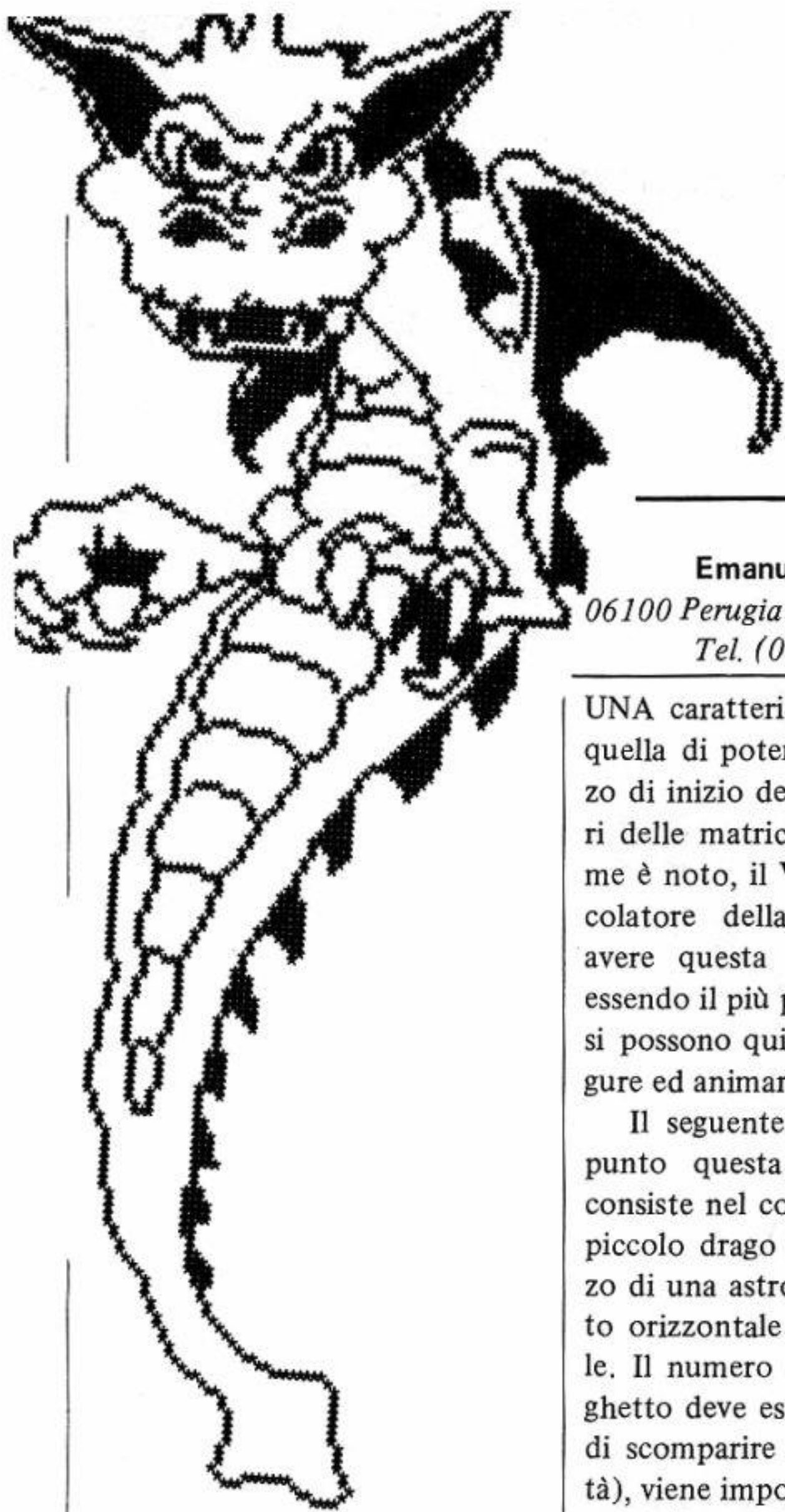
Caratteristiche	Commodore 64	Apple II +	IBM	TRS-80 III	Atari 800
Prezzo base Usa (Commodore = 100)	100	257	263	167	151
Memoria	64K	48K	16K	16K	16K
Tastiera	66 tasti	52 tasti	83 tasti	65 tasti	61 tasti
Caratteri grafici da tastiera	SI	NO	NO	NO	SI
Maiuscole e minuscole	SI	Solo maiuscole	SI	SI	SI
Capacità di memoria dei dischetti da 5" 1/4 (per unità collegata)	170K/1000K	143K	160K	178K	96K
Generatore di suoni	SI	SI	SI	NO	SI
Sintetizzatore di musica	SI	NO	NO	NO	NO
Uscita Hi-Fi	SI	NO	NO	NO	NO
Uscita video	SI	EXTRA	EXTRA	NO	SI
Periferiche "intelligenti"	SI	NO	NO	NO	SI
Disponibilità del CP/M	SI	SI	SI	SI	NO
Cartucce per giochi	SI	NO	NO	NO	SI
Controllori per giochi	SI	SI	SI	NO	SI

di inizio o per cancellare tutti i caratteri dallo schermo; il tasto INST/DEL, per inserire o cancellare i caratteri e correggerli direttamente sullo schermo.

• *Il video* è nel formato 25 linee x 40 colonne, permette di ottenere 255 combinazioni di colori fra traccia grafica o caratteri e sottofondo, è in grado di produrre fino a 16 colori ed usa tutti i caratteri grafici del PET, oltre a consentire all'utente di costruire propri caratteri programmabili in sostituzione di quelli standard.

La grafica può essere senz'altro definita ad alta risoluzione grazie ai suoi 320x200 pixel, che offrono risultati sorprendenti specialmente nei giochi o nelle animazioni.

I suoni sono prodotti dall'interfaccia sonora 6581, in grado di produrre musica e suoni ad un livello professionale, in particolare: tre voci indipendenti, ciascuna con un range di nove ottave, quattro forme d'onda (dente di sega, triangolo, impulso variabile e disturbo) e un filtro programmabile selezionato singolarmente per ogni voce.

A completare le qualità del 64, sono disponibili numerose periferiche che vale la pena di menzionare: il registratore a cassette "Datassette" che utilizza cartucce a nastro standard; l'unità a disco singolo (con propri microprocessori e sistemi di memoria) per minifloppy da 5 pollici, in grado di memorizzare fino a 170.000 caratteri; la stampante VIC 1515, che si collega al 64 direttamente senza interfaccia supplementare ed ha una velocità di stampa di 30 caratteri al secondo.

Sono previste inoltre cartucce per l'interfacciamento del 64 con diversi dispositivi standard quali stampanti, unità di controllo e modem. Fra queste: l'interfaccia IEEE-488, l'emulatore PET, i collegamenti audio e video e gli alloggiamenti cartuccia per giochi e altre applicazioni. ▪

Tiro al drago

Emanuele Morelli
06100 Perugia Via Lorenzini, 22
Tel. (075) 41119

UNA caratteristica del VIC 20 è quella di poter spostare l'indirizzo di inizio della lettura dei valori delle matrici dei caratteri. Come è noto, il VIC è il primo calcolatore della Commodore ad avere questa caratteristica, pur essendo il più piccolo. Con il VIC si possono quindi creare delle figure ed animarle.

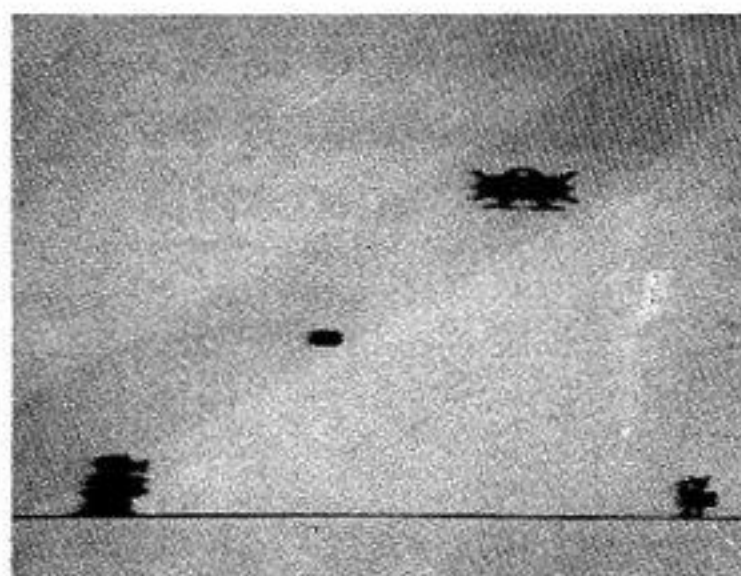
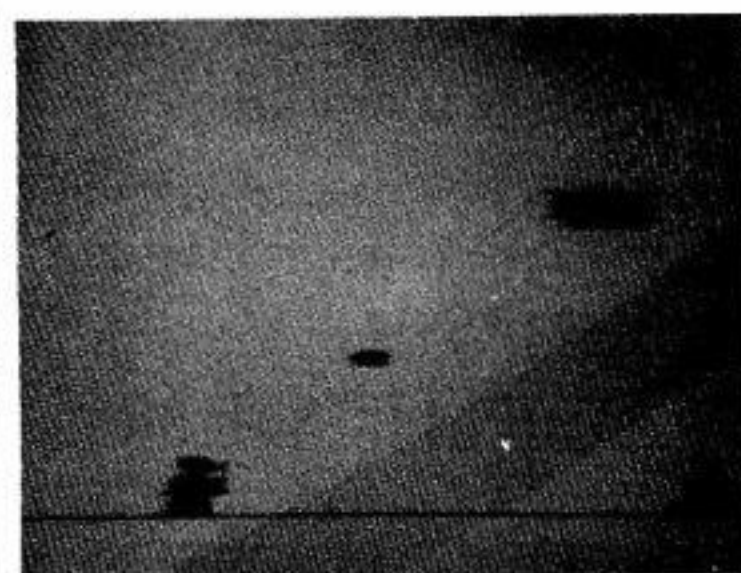
Il seguente gioco sfrutta appunto questa possibilità. Esso consiste nel colpire la testa di un piccolo drago stilizzato per mezzo di una astronave in movimento orizzontale ad altezza variabile. Il numero di volte che il draghetto deve essere colpito, prima di scomparire (grado di difficoltà), viene impostato dal giocatore su richiesta del programma. Per colpire il draghetto si devono usare i due tasti CRSR ↗ ↘ e CRSR → ←: il primo per lanciare obliquamente il proiettile verso sinistra, il secondo per mirare verso destra.

Ad ogni passaggio della astronave il draghetto avanza di un passo sino a raggiungere il piccolo astronauta sulla sinistra. Quando il draghetto è stato colpito un numero di volte pari al grado di difficoltà impostato inizialmente, scompare. Al suo posto se ne riforma un altro. Questo ripartirà dalla posizione iniziale, ma avrà un colore differente.

In ordine si formeranno un draghetto nero, verde e infine blù. Per vincere la partita bisogna far scomparire tutti e tre i draghetti prima che raggiungano l'astronauta che finirebbe così incenerito. Per ricominciare si può premere la barra lunga in basso che serve per creare gli spazi.

Si può notare che il gioco, dopo il run, non parte subito, perchè il programma deve trasferire i caratteri diretti dalla ROM alla RAM (dal byte 6144). Viene poi chiesto il grado di difficoltà che si vuole avere, che deve essere un numero compreso tra I e 16. Subito dopo parte il gioco.

Per mancanza di spazio in memoria non vi sono scritte sullo schermo, nè REM sul listato. Il programma gira sul VIC senza espansioni, cioè con 3583 bytes liberi che il programma sfrutta sino in fondo.

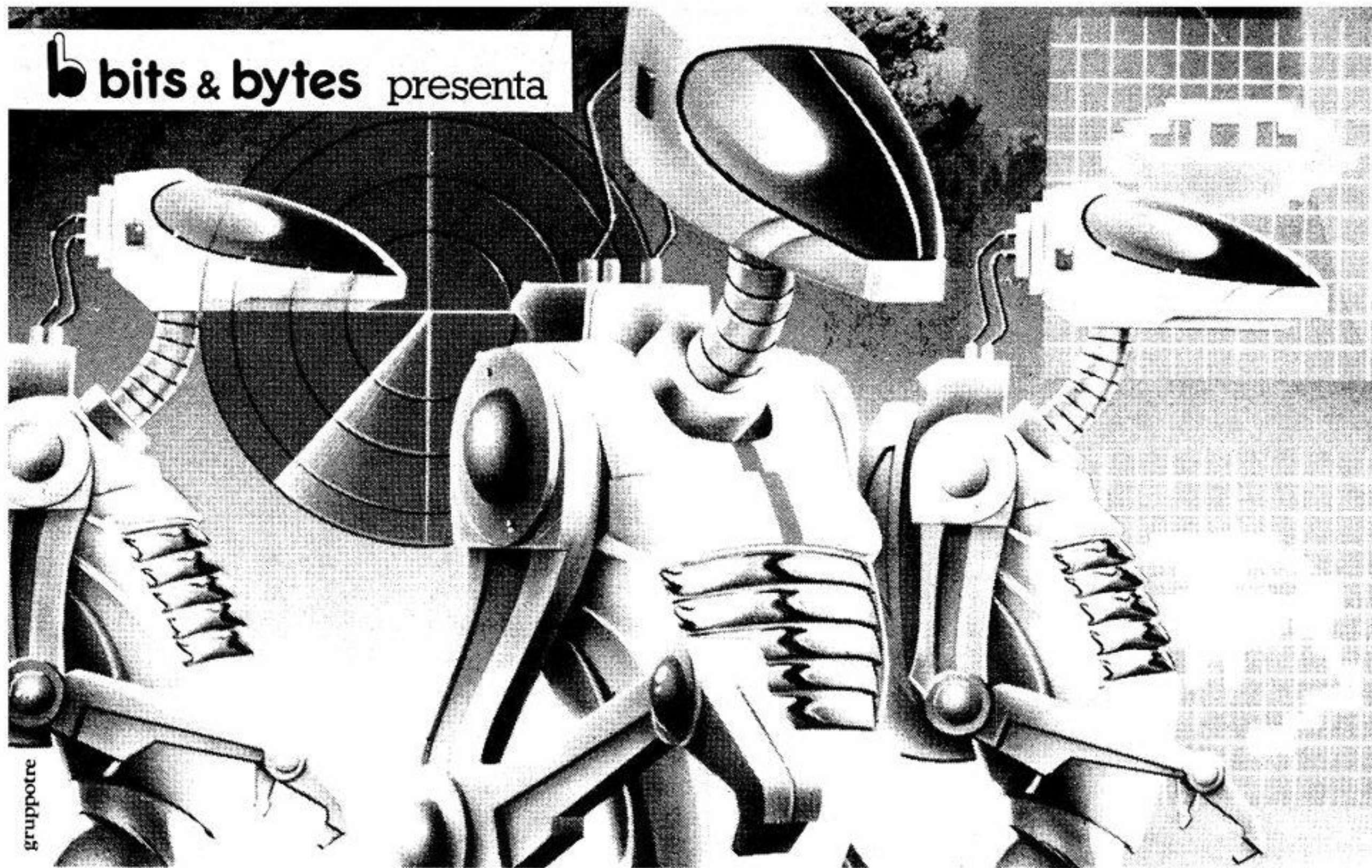
Chi vuole utilizzare il programma deve fare attenzione a non introdurre spazi tra le istruzioni o può finire la memoria e quindi avere una scritta "out of memory error". Per facilità di lettura, comunque, sul listato gli spazi sono stati lasciati.

Descrizione del listato

10-15	Il puntatore interno porta l'inizio della memorizzazione delle stringhe e la fine della memoria al byte 6143.
20-25	Memorizza i caratteri diretti da 6144.
30-34	Memorizza dei nuovi caratteri al posto di quelli in reverse.
50-199	Valori per la definizione di nuove matrici dei caratteri.
200-210	Sposta l'inizio dell'indirizzo di lettura dei caratteri da 32768 (ROM) a 6144 (RAM).
240	Chiede il numero di volte che deve essere colpito (sulla testa) il "draghetto".
300-320	Disegna lo sfondo e l'extraterrestre.
340-345	Disegna il "draghetto".
410-430	Fa partire l'astronave ad altezza casuale.
440-450	Premendo il tasto CRSR right il colpo và verso destra, mentre con CRSR down và a sinistra.
1100-1105	Calcola la posizione del colpo.
1120-1122	Se il colpo raggiunge la testa del "draghetto" il programma prosegue con la linea 2000 altrimenti và a 3000.
1125	Fa permanere l'immagine del colpo per un po' e la cancella.
1152-1155	Cancella l'astronave.
1156-1158	Cancella il "draghetto".
2000	Cancella il colpo.
2005-2010	Il generatore di rumore viene attivato.
2500-2600	Cancella il "draghetto" e lo fa riformare in un nuovo colore.
5000-6000	Discesa dell'astronave e rumore del "draghetto".

Variabili principali

aS	Contiene i valori per la formazione dei nuovi caratteri.
n	Varia da 19 a 2 ed è la distanza del draghetto dall'astronauta.
z	Contiene il numero di volte che è colpito il draghetto.
x	E' il numero di volte che deve essere colpito ogni draghetto.
c	Controlla il colore dei draghetti.
m	E' casuale: altezza dell'astronave.
b	Controlla la traiettoria del colpo.
p	Posizione orizzontale dell'astronave.
g	Posizione del colpo.
l	Ha valore 0 quando il colpo non è partito ed 1 quando lo è.
s	Ha valore 1 quando il draghetto viene raggiunto x volte dal colpo.
ff	Diviene = 1 dopo aver fatto scomparire il terzo draghetto.

```
5 PRINT"";
10 POKE51,255:POKE52,23
11 POKE55,255:POKE56,23
20 FORI=0TO1023
21 POKE6144+I,PEEK(32768+I):NEXT
30 FORI=0TO1023
31 READA$
32 IFA$="*"THEN200
33 POKE7168+I,VAL(A$)
34 NEXT
50 DATA0,5,7,141,95,199,95,143
60 DATA0,64,208,224,240,224,192,192
70 DATA3,103,231,127,63,79,255,255
80 DATA200,248,240,244,252,248,250,254
90 DATA63,7,15,15,15,12,12,31
100 DATA252,248,248,252,252,252,206,199
110 DATA68,40,62,49,49,31,222,254
120 DATA254,254,254,254,18,18,18,27
132 DATA0,128,64,39,31,29,63,127
134 DATA60,66,129,255,255,90,255,255
```

```
136 DATA0,1,2,228,248,184,252,254
138 DATA63,31,47,71,128,1,2,31
140 DATA0,255,255,255,129,0,0,195
142 DATA252,248,244,226,1,128,64,248
199 DATA*
200 POKE36869,254
210 POKE36866,PEEK(36866)OR128
240 PRINT"J";:INPUT"DIFFICOLTA' (1-16)";X
300 PRINT"J";:FORI=1TO21:PRINT"X";:NEXT:OF=30720
310 POKE36879,25:FORI=1TO22:PRINT"‾";:NEXT:FORH=38444TO38774:POKEH,0:NEXT
320 POKE8099,134:POKE8121,135:POKE38819,2:POKE38841,2
330 PRINT"■";
335 N=19:PRINT"TTTJ";
340 PRINTTAB(N)"■@A"
342 PRINTTAB(N)"■BC"
344 PRINTTAB(N)"■DE"
345 PRINT"TTJ";
400 IFN=2THEN6000
410 M=INT(RND(1)*10)*22+7768
420 FORP=0TO19:POKEM+P,136:POKEM+P+1,137:POKEM+P+2,138:POKEM+P+22,139
430 POKEM+P+23,140:POKEM+P+24,141:FORT=1TO140:NEXT
440 IFL=0ANDPEEK(197)=31THENL=1:B=21:GOTO1100
450 IFL=0ANDPEEK(197)=23THENL=1:B=23:GOTO1100
480 GOTO1150
1100 E=1:G=M+45+P+B*F:POKEG,81:POKEG+OF,0
1105 IF((G/22-INT(G/22))*22-2)<.001THEN3000
1120 IFB=21AND(PEEK(G+21)=128ORPEEK(G+21)=129)THEN2000
1121 IFB=23AND(PEEK(G+23)=128ORPEEK(G+23)=129)THEN2000
1122 IFPEEK(G+21)<>32ORPEEK(G+22)<>32ORPEEK(G+23)<>32THEN3000
1125 FORT=1TO10:NEXTT:POKEG,32:F=F+1:E=0:GOTO1100
1150 IFS=1THEN2500
1152 POKEM+P,32:POKEM+P+22,32:NEXTP:N=N-1:POKEM+P,32:POKEM+P+22,32:POKEM+P+1,32:
POKEM+P+23,32
1154 IFFF=1THEN5000
1155 POKEM+P+2,32:POKEM+P+24,32:L=0
1156 PRINTTAB(N)"   "
1157 PRINTTAB(N)"   "
1158 PRINTTAB(N)"   ":PRINT"TTJ";
1159 IFS=1THEN2500
1160 GOTO340
2000 POKEG,32:Z=Z+1
2005 FORV=15TO0STEP-1:POKE36878,V:POKE36877,140+Z*7:FORT=1TO200:NEXT:F=0
2010 POKE36874,150:NEXTV
2020 IFZ=XTHENS=1:Z=0:GOTO1150
2030 GOTO1150
2500 S=0:FORU=1TO3:PRINTTAB(N)"   ":NEXTU:PRINT"TTJ";:C=C+1:N=19:IFC=3THENFF=1:GO
TO1150
2600 PRINTCHR$(29+C);
3000 POKEG,32:F=0:E=0:GOTO1150
5000 PRINT"XⅢ":PRINTTAB(4)"■HIJ■■■■KLM■":GOTO6200
6000 POKE36878,15:POKE36877,130:FORT=1TO3000:NEXT:POKE36878,0
6100 POKE38841,0:POKE38819,0
6200 IFPEEK(197)=32THENRUN240
6300 GOTO6200
```

I NUOVI PROGRAMMI PER IL VIC 20®
DELLA UNITED MICROWARE INDUSTRIES INC.

VUM01N VICALC
Un comodo programma di calcolo: i suoi 10 registri di memoria e 4 di dati sullo stack sono costantemente visualizzati sullo schermo. Premendo un singolo tasto, potrete ottenere i risultati di funzioni aritmetiche e matematiche, percentuali, numeri casuali, interessi composti e così via. Questi risultati possono essere immagazzinati in qualunque registro di memoria. La precisione può essere scelta tra 0 e 9 cifre significative. Lit. 36.000 + IVA 18%.

VUM02N ALIEN BLITZ
Un gioco d'azione in cui dovrete liberare il cielo da invasori alieni. Richiede riflessi prontissimi, specialmente ai livelli superiori di difficoltà. Comprende diversi «specials» per chi raggiunge alti punteggi o colpisce determinati obiettivi. Joystick. Lit. 60.000 + 18%.

VUM04N THE ALIEN 6K WITH JOYSTICK
Questa volta l'alieno siete voi, e dovrete difendervi, con trappole aeree gonfiabili ed altri espedienti, dagli abitanti di Critter City che cercano di annientarvi. Azione rapidissima ed effetti sonori. Espansione 3K. Lit. 60.000 + IVA 18%.

VUM06N 3-D MAZE
Dovrete trovare la vostra strada attraverso le pareti ed i corridoi di un intricato labirinto. I vari passaggi si aprono realisticamente ai vostri occhi mentre cercate affannosamente una via d'uscita. Vi intratterrà per ore ed ore. Lit. 36.000 + IVA 18%!

VUM09N VICAT
Il vostro data base personale: questo programma puo gestire elenchi di nomi, indirizzi, numeri telefonici, ricette e molte altre cose. I records possono essere modificati, alfabetizzati, registrati o prelevati da nastro. Vicat è particolarmente potente quando i suoi primi files vengono usati come directories. Ciascuno dei files (45 linee con 17 caratteri per linea) possono essere analizzati con una rapida funzione di ricerca. Espansione 3K o 8K. Lit. 60.000 + IVA 18%.

VUM10N AMOK
I corridoi di Amok sono popolati da robots la cui sola finalità consiste nell'annientare i visitatori indesiderati. Per salvarvi, dovrete prendere decisioni rapide e muovervi velocemente. I molti livelli di difficoltà di questo gioco vi daranno ore di divertimento. Joystick. Lit. 60.000 + IVA 18%.

VUM11N VITERM A
Un programma che vi consentirà di connettervi telefonicamente con reti di comunicazione e di trasformare il vostro Vic in un terminale di computers più grandi. Richiede un modem con interfaccia RS-232. Lit. 48.000 + IVA 18%.

VUM12N VICHECK
Un programma per gestire la vostra contabilità bancaria, compresi assegni, depositi e tutte le normali operazioni. Funzioni di controllo, di ricerca rapida, di aggiornamento e perfino di stampa assegni. Espansione 3K o 8K. Lit. 60.000 + IVA 18%.

VUM 12 N SUB CHASE
Una flotta di sommergibili sgancia mortali mine ad immersione. Per distruggerle, dovrete piazzare al posto giusto delle barriere di bombe di profondità. Più riuscirete a distruggerne, tuttavia, più l'attacco nemico si farà serrato! Espansione 8K. Lit. 60.000 + IVA 18%.

RICHIEDETELI AL VOSTRO RIVENDITORE O DIRETTAMENTE A **bits & bytes** 20124 Milano

Via Monte Grappa n. 4 Tel. 02/573344-5468901-651628

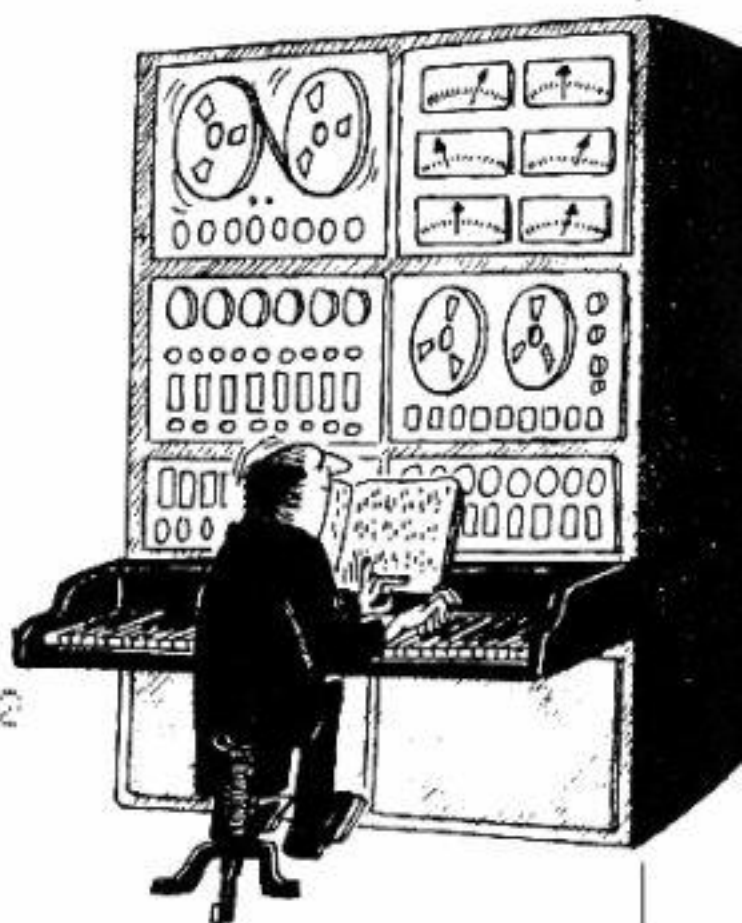
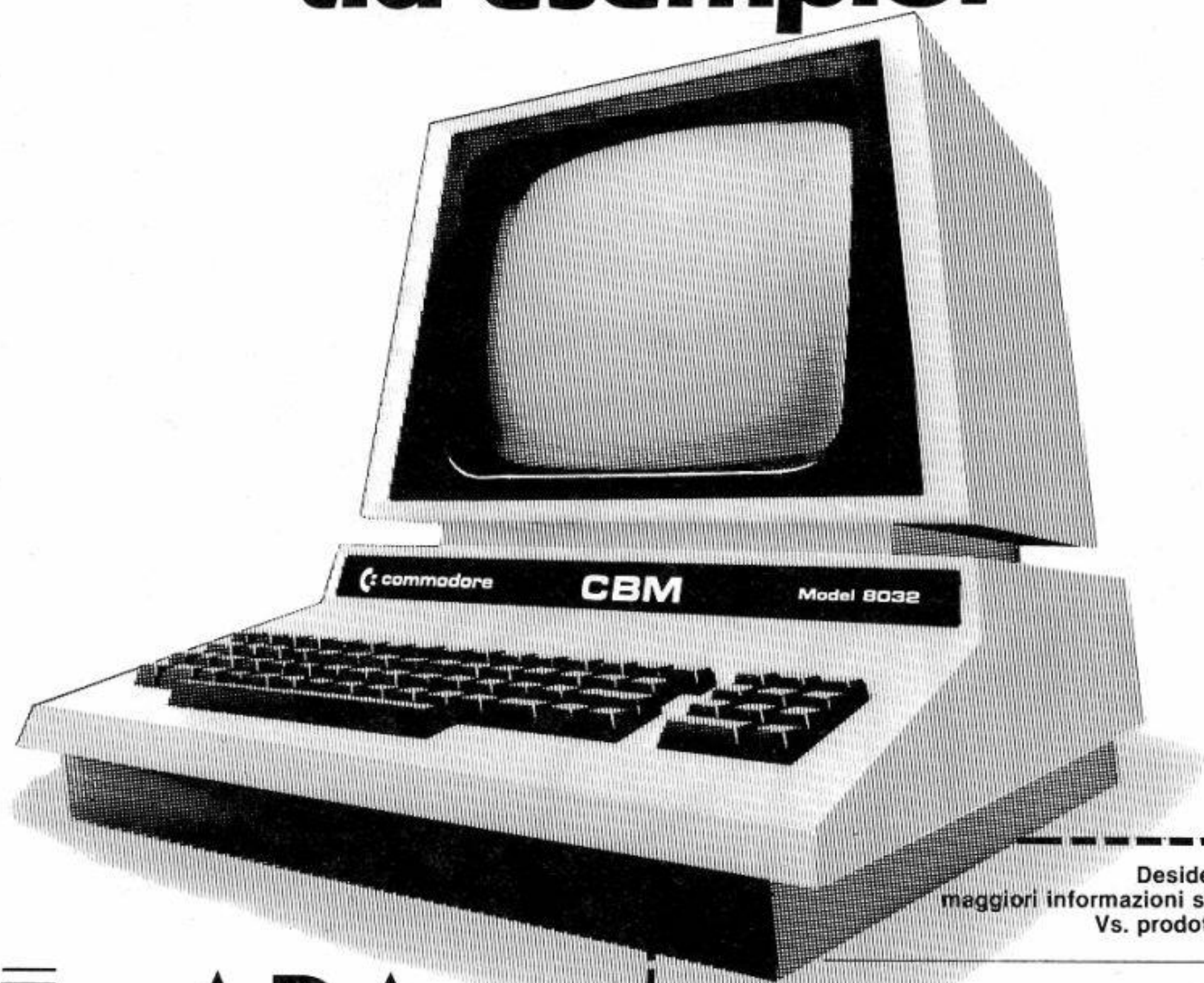
bits & bytes tutto il software che volete

Tastiera musicale

QUESTO programma, intitolato "Sintetizzatore", sfrutta i registri del suono. La sua particolarità è di mantenere la nota impostata (una per ogni tasto della tastiera) sino a che non viene staccato il dito, proprio come in un organo; le note sono suonate con la stessa sequenza dei tasti sulla tastiera (sinistra - destra e alto basso). I 4 timbri possono essere impostati in qualsiasi momento premendo i 4 tasti FI-F7 sulla destra. Inoltre possono essere sovrapposti i timbri stessi.

Il programma gira sul VIC in configurazione standard. La cartuccia per alta risoluzione, non permette una cosa simile (occorre impostare il tempo della nota).

Emanuele Morello

```
1 PRINT"■■";
2 PRINT"■    SINTETIZZATORE    ■"
4 PRINT"■TIMBRI F1-F7":PRINT
7 PRINT"■CRSR ← NOTA SOMMATA   CONTINUA":PRINT:PRINT"CRSR ↑ NOTA SOMMATA   NON C
ONTINUA"
8 X=36876:V=15:P=15
9 PRINT:PRINT"SPACE AZZERA E PORTA  AL TERZO GENERATORE"
10 Z=PEEK(197)
12 IFZ=64THENPOKEX,0:POKE36878,P:GOTO10
18 POKE36878,V:IFV<1THENV=1
19 IFV>15THENV=15
20 IFZ>22ANDZ=<44THENA=Z-22:GOTO35
25 IFZ>44THENB=Z-44:GOTO40
30 ONZGOTO52,54,56,58,60,62,64,10 ,66,68,70,72,74,76,10,10,78,80,82,84,86,88
35 ONAGOTO106,10,10,91,93,95,97,99,111,110,90,92,94,96,98,10,102,10,79,81,83,85
40 ONBGOTO87,89,103,65,67,69,71,73,75,77,104,51,53,55,57,59,61,63,105
51 POKEX,135:GOTO10
52 POKEX,143:GOTO10
53 POKEX,147:GOTO10
54 POKEX,151:GOTO10
55 POKEX,155:GOTO10
56 POKEX,159:GOTO10
57 POKEX,163:GOTO10
58 POKEX,167:GOTO10
59 POKEX,175:GOTO10
60 POKEX,179:GOTO10
61 POKEX,183:GOTO10
62 POKEX,187:GOTO10
63 POKEX,189:GOTO10
64 POKEX,191:GOTO10
65 POKEX,201:GOTO10
66 POKEX,203:GOTO10
67 POKEX,207:GOTO10
68 POKEX,209:GOTO10
69 POKEX,212:GOTO10
70 POKEX,215:GOTO10
71 POKEX,217:GOTO10
72 POKEX,219:GOTO10
73 POKEX,221:GOTO10
74 POKEX,223:GOTO10
75 POKEX,225:GOTO10
76 POKEX,227:GOTO10
77 POKEX,229:GOTO10
78 POKEX,229:GOTO10
79 POKEX,231:GOTO10
80 POKEX,232:GOTO10
81 POKEX,233:GOTO10
82 POKEX,235:GOTO10
83 POKEX,236:GOTO10
84 POKEX,237:GOTO10
85 POKEX,238:GOTO10
86 POKEX,239:GOTO10
87 POKEX,240:GOTO10
88 POKEX,241:GOTO10
89 POKEX,243:GOTO10
90 POKEX,244:GOTO10
91 POKEX,245:GOTO10
92 POKEX,246:GOTO10
93 POKEX,247:GOTO10
94 POKEX,248:GOTO10
95 POKEX,249:GOTO10
96 POKEX,250:GOTO10
97 POKEX,251:GOTO10
98 POKEX,252:GOTO10
99 POKEX,253:GOTO10
101 POKEX,254:GOTO10
102 X=36874:GOTO10
103 X=36875:GOTO10
104 X=36876:GOTO10
105 X=36877:GOTO10
106 P=15:GOTO10
110 FORI=1TO4:POKE36873+I,0:NEXTI:GOTO10
111 P=0:GOTO10
```

CON IL BINOMIO ATLAS - COMMODORE IL COMPUTER NON È PIÙ UN MITO MA UNA REALTÀ ALLA PORTATA DI TUTTI
ATLAS
system s.r.l.
Se la qualità del servizio, il giusto costo l'affidabilità delle macchine e dei programmi costituiscono per Voi scelte primarie e peso decisionale, l'interazione fra le Vostre aspirazioni e la nostra professionalità possono coesistere e stimolarVi ad interpellarci per le Vostre esigenze.
ATLAS Vi fornisce programmi standard o personalizzati, gestionali o scientifici, provvede anche all'addestramento dell'operatore e Vi consiglia sulla forma di acquisto che meglio si confà alle Vostre risorse.
Studio di:
INFORMATICA-PROGETTAZIONE-MARKETING
ATLAS SYSTEM SRL Via Marconi, 17
01100 VITERBO Tel. 0761/224688

IMPARA A PROGRAMMARE CON IL VIC

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ DISPENSA N.1 ★

Commodore Computer per tutti, per tutti gli usi, per tutte le tasche.

Affari o amministrazione, industria o problemi scientifici, agricoltura, didattico o divertimento.
C'è un Commodore Computer per tutti. Dal fantastico Vic 20 alla sofisticatissima serie 8000, i computer a basso costo ed alta capacità per l'elaborazione dati e il word processing.

Commodore Computer è il n. 1 dei personal in Europa, fra le prime tre aziende nel mondo. Grazie alla alta qualità dei suoi prodotti. Fai un salto da un Rivenditore autorizzato Commodore:
un mondo di buone idee ti aspetta!

commodore COMPUTER

Commodore Italiana srl - Milano via Conservatorio 22, tel. 74.91.126

Sono interessato a ricevere materiale illustrativo Commodore relativamente a questi settori:

- ☐ Gestionale
- ☐ Scientifico
- ☐ Divertimento/didattica

nome____________________

indirizzo____________________

8

Premessa

Le meravigliose caratteristiche del computer VIC Colour della Commodore vi consentono di utilizzarlo per una vasta gamma di applicazioni in casa, a scuola, negli uffici e laboratori, insomma ovunque possiate attaccarlo ad una presa di corrente e ad un televisore.

Non appena acceso il vostro VIC, vi dà l'accesso al linguaggio di programmazione BASIC, il cui nome deriva dalle iniziali inglesi di Beginners All-purpose Symbolic Instruction Code, che vuol dire serie codificata di istruzioni simboliche per qualsiasi utilizzazione da parte di utenti non specializzati.

Imparare a scrivere utili programmi in basic è facile e questo libro ve lo proverà. Non è necessario che già conosciate i computer. La lettura del volume ed i numerosi esercizi e problemi (le soluzioni sono riportate in appendice ai vari capitoli) vi aiuteranno a progredire secondo i vostri normali ritmi d'apprendimento ed il tempo che potrete dedicargli.

Una caratteristica particolare del VIC è l'utilizzazione del colore del vostro TV. Il volume, perciò, vi insegnerà ad utilizzare fino dall'inizio il colore e una serie di semplici tecniche d'animazione, affinchè possiate introdurre tali caratteristiche nei programmi che andrete sviluppando.

Il VIC vi consente altresì di programmare una serie di effetti musicali. Nel libro troverete delle routine esemplificative perchè possiate inserirle nei vostri programmi.

La versatilità del VIC è dimostrata ulteriormente dalla sezione dedicata alla grafica con alta risoluzione. Questa non richiede programmi speciali, ma soltanto un modulo di ampliamento della memoria da 3 mila caratteri (3K RAM). Tutte le subroutine necessarie per la grafica ad alta risoluzione sono riprodotte in questo volume.

Altre applicazioni di cui parleremo sono:

- effetti eromatici, animazione e grafica ad alta risoluzione;
- effetti musicali e trasmissione Morse;
- calcoli dei costi e dei mutui;
- elaborazione e simulazione della gestione del portafoglio titoli;
- elaborazione di dati scientifici ed ingegneristici;
- calcoli matematici di varia complessità.

Ovviamente questi esempi non esauriscono i campi di applicazione del VIC. Ma una volta che grazie ad essi avrete imparato a padroneggiare il VIC, sarete in grado di passare a lavori più complessi e sofisticati. Ad esempio alla parte utente del VIC la cui utilizzazione può consentirvi di controllare modelli ed apparecchiature scientifiche.

Nel mercato sono disponibili diverse unità accessorie per il VIC, come la clôche (in inglese joy-stick) e la "paletta" (paddle), utili nei giochi e nelle applicazioni di controllo, un'unità di memoria su dischetto ed una stampante per più avanzate applicazioni gestionali.

1. Approccio ai computer ed alla programmazione

Come è fatto e cosa fa un computer

Una caratteristica fondamentale del computer è la sua capacità di tenere in memoria la serie di istruzioni necessarie per svolgere qualsiasi elaborazione. Questo insieme di istruzioni (il *programma),* è preparato dal programmatore che deve trovarsi nella *memoria* principale del computer, perchè i vari ordini possano essere eseguiti.

Ciascun computer ha un insieme fisso di istruzioni che è in grado di eseguire. *L'unità di controllo* richiama le istruzioni della memoria una per volta, le decodifica o interpreta e fa si che il computer le esegua. Se l'istruzione richiede un'operazione aritmetica, l'unità di controllo trasferisce i dati necessari dalla memoria all'*unità artimetica e logica.*

La memoria centrale, l'unità logica e l'unità di controllo costituiscono il cuore del computer. Insieme sono noti come *elaboratore centrale* (central processor).

Le apparecchiature *periferiche di ingresso (input)* e *di uscita (output),* collegate all'elaboratore centrale, servono per immettere (input) i programmi e i dati nella memoria del computer e per ottenere in uscita (output) i risultati delle elaborazioni.

Le unità di input tipiche leggono e decodificano il significato dei fori perforati sulle classiche schede formato dollaro o su nastro di carta, oppure "sentono" i segnali ottici o magnetici dei supporti appositi e trasmettono elettronicamente queste informazioni all'unità di elaborazione centrale.

In alternativa l'informazione può essere digitata direttamente da una tastiera (simile a quella delle macchine da scrivere).

Il risultato può venire visualizzato sullo schermo d'un televisore, d'un monitor o di altri tipi di display. Se occorrono copie documentali (hard = dure, rigide) le stampanti in commercio possono riprodurre i caratteri uno per volta, una riga alla volta, su foglio o su carta continua. Per riprodurre dei disegni si usano dei tracciatori grafici speciali (plotter).

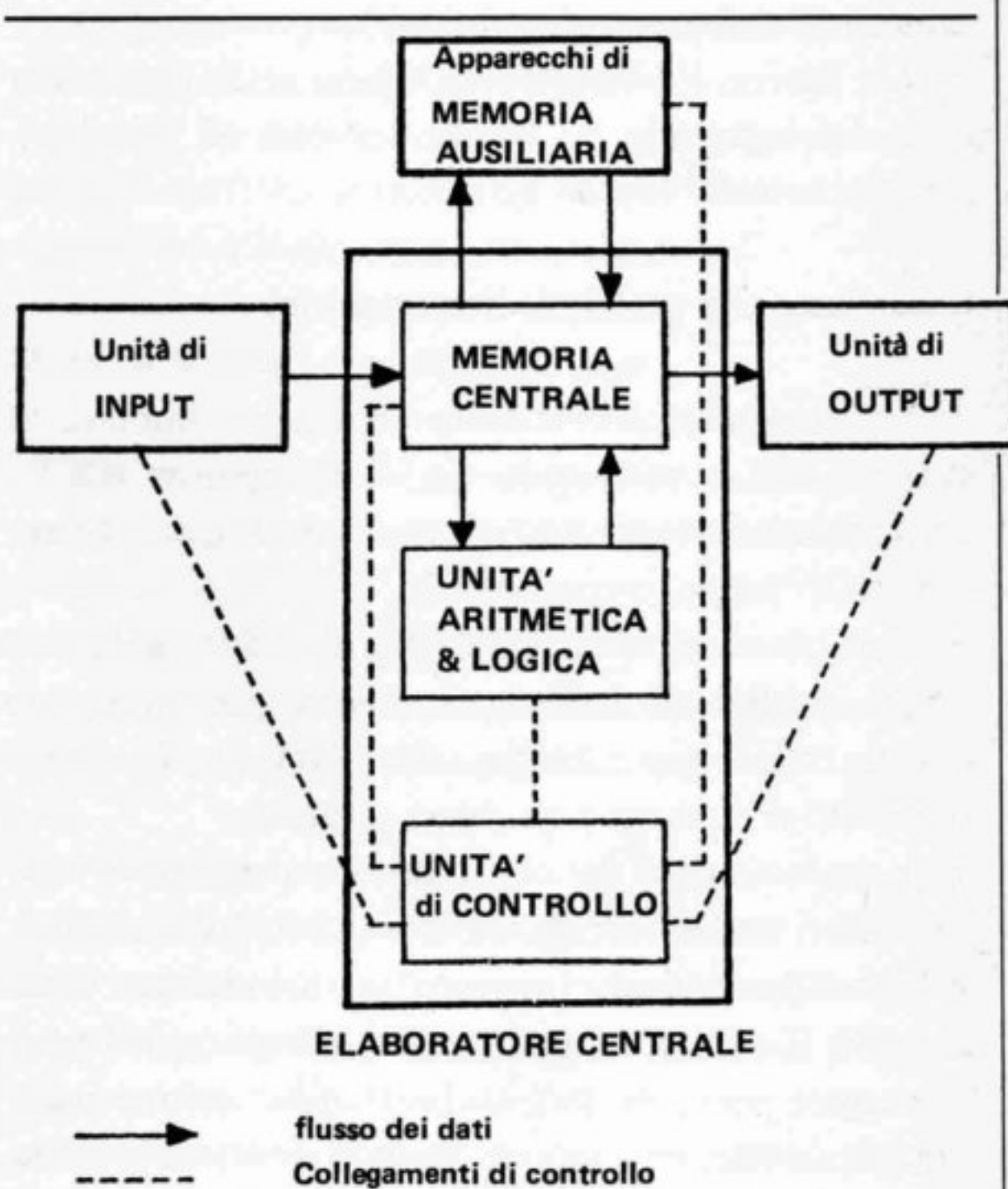



Fig. 1.1. Elementi costitutivi di un computer

I programmi possono essere memorizzati magneticamente su apparecchiature di *memoria secondaria, o di massa,* per poi farli rileggere e trasferire nella memoria del computer all'occorrenza. Nei grandi computer (detti anche mainframe) a questo scopo si usano dischi e nastri magnetici. Nei più piccoli microcomputer si ricorre ai dischetti flessibili (floppy disk) ed alle cassette che riescono a contenere meno informazioni e le trasferiscono

più lentamente. Ma sono disponibili altresì altri tipi di memorie ausiliarie come quelle a bolle magnetiche.

Le unità di memoria di massa servono altresì ad immagazzinare gli archivi (file) dei dati registrati. Questi vengono trasferiti dalla memoria del computer sotto il controllo del programma per le applicazioni di elaborazione degli archivi.

La figura 1.1 mostra gli elementi costitutivi di base del computer, il flusso dei dati ed i collegamenti di controllo.

Come vengono gestite le informazioni

Un computer è costituito da un gran numero di componenti a due stati. Lo stato spento (OFF) convenzionalmente rappresenta uno 0. Lo stato acceso (ON) rappresenta un 1.

Un sistema di numerazione che utilizza solo lo 0 e l'1 è detto *binario.* Combinando opportunamente queste cifre binarie (dette anche *bit*) si può ottenere qualsiasi numero e qualsiasi carattere.

Nella memoria del computer i numeri sono rappresentati da una combinazione di bit. La quantità di bit utilizzata per rappresentare un numero varia secondo il computer usato. Con il linguaggio basic, la maggior parte dei sistemi lavorano in aritmetica a virgola mobile, ed i numeri sono memorizzati come *montissa ed esponente.* Ad esempio, $6x10^3$ = 6.000 ha una montissa 6 e come esponente 3.

La programmazione di un computer

Ogni famiglia di processori ha un proprio insieme di istruzioni, normalmente diverso da quello di tutti gli altri. Ciò significa che ogni dato processore è capace di capire soltanto il suo esclusivo insieme di istruzioni in *codice binario.*

La memoria del computer può essere immaginata come un alveare di celle capaci di immagazzinare le combinazioni binarie rappresentanti le istruzioni del programma ed i dati. Ciascuna cella è contraddistinta da un numero esclusivo grazie al quale è possibile indirizzarsi in modo inequivocabile ad ognuna di esse per scriverci dentro dei dati o per leggervi le informazioni e le istruzioni in essa memorizzate.

Tabella 1.1 Istruzioni in linguaggio macchina

Istruzione	*Linguaggio macchina*
1 Carica il numero della cella 5 nell'accumulatore	10100101 0101
2 Aggiungi il numero della cella 6 a quello dell'accumulatore	01100101 0110
3 Memorizza il numero dell'accumulatore nella cella 8	10000101 1000

Per dare un esempio di come i programmi sono scritti nel codice proprio di un computer *(linguaggio macchina)* immaginiamo due numeri memorizzati nelle celle 5 e 6. Questi due numeri possono essere addizionati ed il risultato può venire a sua volta conservato, mettiamo, nella cella 8. L'addizione richiesta verrà effettuata in una particolare localizzazione di memoria, detta *accumulatore.* La prima istruzione del programma, pertanto, comanderà di caricare uno dei due numeri nell'accumulatore. La seconda istruzione richiederà di aggiungere il secondo numero a quello precedentemente inviato nell'accumulatore il quale, a questo punto, si troverà a memorizzare la somma dei due numeri. La terza istruzione mira a trasferire e memorizzare il risultato contenuto nell'accumulatore nella cella di memoria voluta.

La codifica binaria di queste istruzioni espressa per un processore tipo è esposta nella tabella 1.1

Se nelle celle 5 e 6 avessimo memorizzato rispettivamente i numeri 70 e 25, dopo l'esecuzione delle tre istruzioni della tabella 1.1, le celle 5 e 6 conterrebbero ancora i numeri 70 e 25, mentre nella cella 8 troveremmo 70 + 25 ossia 95. Lo stesso programma può essere riutilizzato cambiando i dati delle celle 5 e 6 (mettiamo 43 e 12), col risultato che nella cella 8 il precedente valore 95 verrebbe sostituito dal nuovo valore 55.

I linguaggi di programmazione

Come abbiamo visto, per programmare in linguaggio macchina occorre dare le istruzioni ed i dati in binario. Ma scrivere e battere sulla tastiera tutti questi 1 e 0 richiede troppo tempo e può dare luogo a tanti errori. Per eliminare questi inconvenienti si ricorre a dei *codici mnemonici.* Ad esempio, il comando di caricare un numero da una cella di memoria potrebbe essere scritto come LOAD

(dall'inglese load = carica) era invece, mettiamo, di 10100101. Analogamente alle celle di memoria potremmo attribuire dei nomi simbolici invece di indicarle con i loro veri numeri di indirizzo in binario.

Questo tipo di linguaggio di programmazione aiuta a controllare pienamente le funzioni del computer. I linguaggi che utilizzano simili codici mnemonici sono chiamati *linguaggi assemblatori.* Ogni istruzione del linguaggio assemblatore normalmente corrisponde ad una equivalente istruzione in codice macchina. La traduzione di un programma dal linguaggio assemblatore al linguaggio macchina viene eseguita da un particolare programma in codice macchina detto *assemblatore.*

Una particolare categoria è quella dei linguaggi ad alto livello la cui caratteristica prima consiste nell'esprimere con un solo comando tutta una serie di istruzioni in linguaggio macchina. Il *basic* è uno di essi. Vediamo ad esempio la seguente espressione:

LET C = A + B

dove LET è un termine inglese corrispondente al nostro "sia". Questa istruzione basic fa sì che i contenuti delle celle di memoria A e B siano addizionati e la somma venga conservata nella cella C. Questo stesso problema in precedenza (tabella 1.1) aveva richiesto tre istruzioni in codice macchina o in linguaggio assemblatore.

Ma nè il linguaggio assemblatore nè i programmi in basic possono venire compresi direttamente dal computer. A tale scopo i programmi basic debbono essere tradotti in linguaggio macchina mediante un *compilatore* o un *interprete.* La differenza tra questi ultimi due consiste nel livello dove avviene la traduzione in codice macchina. Con il compilatore la traduzione avviene *prima* che il programma venga eseguito, cosa che rende il compilatore molto più veloce del traduttore dato che con questo la traduzione viene effettuata man mano che il programma viene eseguito.

L'interprete del VIC BASIC (ossia del basic che gira sul VIC) è particolarmente utile per chi si avvicina per la prima volta al computer, dato che è stato progettato in modo da facilitare al massimo la correzione e la modifica dei programmi.

Microprocessori e microcomputer

Lo sviluppo della microelettronica ha messo il mondo dei computer alla portata di chiunque, modoficando altresì il modo in cui queste macchine venivano utilizzate. Inoltre le ridotte dimensioni ed il basso costo dei sistemi basati su microcomputer rendono possibile l'allargamento dei campi applicativi.

I congegni microelettronici (circuiti integrati) sono costituiti da minuscoli e sottili strati di materiale semiconduttore, come il silicio. In un piccolo quadratino *(chip,* leggi "cip") di silicio della superficie di pochi millimetri quadrati, può contenere un elevatissimo numero di componenti elettronici incorporati nei circuiti.

I circuiti integrati *(IC)* sono disponibili secondo una vasta gamma, in base alle più varie funzioni di memoria e di elaborazione. Oggi è possibile disporre di tutti i circuiti occorrenti per un microcomputer, raggruppati in un unico chip semiconduttore, all'incirca della stessa dimensione dei primi circuiti integrati i quali contenevano solo pochi componenti. I circuiti integrati ad elevata integrazione *(LSI* = large scale integration) contengono diverse migliaia di componenti.

In un microcomputer, il chip che svolge le funzioni di processore centrale è chiamato *microprocessore.* Questo componente svolge in particolare le seguenti funzioni:

- sincronizzazione degli eventi dell'elaborazione e decodifica delle istruzioni (unità di controllo);

• memorizzazione temporanea degli indirizzi e dei dati (registri);
• operazioni aritmetiche e logiche (unità aritmetica e logica).

Il VIC utilizza un microprocessore costruito dalla stessa Commodore, il 6502, ed un processore, sempre della Commodore, per il controllo dell'interfaccia col video.

Quest'ultimo componente è un microprocessore speciale il cui compito consiste nel controllare i segnali sonori e di colore inviati al nostro televisore. Inoltre esso contiene speciali registri di controllo che è possibile modificare con appositi programmi.

2. Semplici istruzioni di ingresso e uscita

I comandi READ, DATA, INPUT e PRINT

In questo capitolo vedremo come immettere *(input)* le istruzioni nel vostro computer e come il VIC può essere programmato per fornire in uscita le informazioni *(output)*, ad esempio visualizzandole sullo schermo o stampandole.

Ciascuna istruzione (o dichiarazione) basic comprende un ordine al computer perchè svolga una data funzione ed una combinazione di variabili, costanti, separatori (come la virgola) ed operatori (ad esempio il segno +) sui quali deve essere esercitato l'ordine impartito. Ecco un esempio:
10 READ A,B,C

Questa istruzione ordina al VIC di leggere (in inglese si dice "read") tre numeri (costanti numeriche) dalla dichiarazione DATA (vedi più in là il rigo di programma 20) e di riportarli in tre celle di memoria del VIC identificate con le lettere A, B e C. Queste tre lettere (A, B e C) sono dette *variabili* e stanno ad indicare i numeri di tre precisi indirizzi di memoria per come si è visto nel capitolo 1.

Tutte le volte che nel corso dello *stesso* programma si farà riferimento ad A,B e C il VIC richiamerà i valori contenuti al momento in queste celle. In un programma diverso, tuttavia, il computer potrebbe assegnare i valori A, B e C a celle con diverso indirizzo, ma la loro identificazione rimarrà invariata per tutto il tempo in cui il nuovo programma starà girando.

In basic le differenti celle di memoria vengono sempre indicate con lettere dell'alfabeto, dalla A alla Z, eventualmente seguite da un numero (da 0 a 9) o da un'altra lettera.

Il 10 che vedete prima dell'ordine READ (leggi) dell'istruzione riportata sopra è il *numero di rigo* (o linea). Fra l'altro questo numero vi consente di modificare tutto il contenuto del rigo ribattendo sulla tastiera il numero stesso e l'istruzione modificata.

Nel numerare le varie linee di istruzione è buona norma lasciare nella loro sequenza numerica degli intervalli (ad esempio numerando da 5 in 5 o da 10 a 10) in modo da avere lo spazio per l'inserimento di eventuali ulteriori istruzioni.

Tre ulteriori istruzioni completano il programma che richiede la lettura e la visualizzazione di tre numeri:
20 DATA 25, 11, 30
30 PRINT A, B, C
40 END

L'istruzione END (fine) conclude l'esecuzione del programma basic, ad esempio fermando l'esecuzione delle sue istruzioni.

Da notare che le tre istruzioni seguenti producono lo stesso risultato del precedente comando READ. In altri termini, dopo l'esecuzione dell'istruzione DATA della linea 20, A, B e C conteranno rispettivamente i valori 25, 11 e 30. Ecco le istruzioni di cui si è detto:

```
10 READ A
11 READ B
12 READ C
20 DATA 25,11,30
30 PRINT A,B,C
40 END
```

Battete questo programma sulla tastiera del vostro VIC esattamente come è, pressando il tasto RETURN (che sta per "a capo") alla fine di ogni rigo. Dopo che avete digitato tutto il programma, battete ancora LIST in modo da poter rivedere ben in ordine e con tutte le correzioni ed aggiunte che avete eventualmente apportato la sequenza del vostro programma e verificatene l'esatezza. Se riscontrate qualche errore, potrete "editare" il programma muovendo il cursore alla posizione desiderata,

utilizzando a tale scopo gli speciali tasti di controllo cursore, grazie ai quali potete muovere il cursore stesso da sinistra verso destra o viceversa, dall'alto verso il basso o viceversa. Una volta raggiunta sullo schermo la posizione dell'errore battete sull'errore stesso i caratteri giusti. Apportata la correzione, battete il tasto RETURN mentre il cursore è sempre sulla linea che state modificando. Quando ridigiterete l'ordine LIST, vedrete apparire sullo schermo la versione modificata del programma.

Per facilitare le operazioni di correzione del testo del programma, esiste uno speciale tasto insert/delete (inserisci/cancella) per l'inserimento di spazi vuoti e la cancellazione di caratteri.

Il comando LIST può essere utilizzato altresì per visualizzare una sola linea di programma (ad esempio la 30) o un gruppo di linee (ad esempio battendo LIST 200-250, comando col quale si chiede di vedere tutte le istruzioni comprese dalla linea 200 alla 250).

Una volta immesso correttamente tutto il programma, digitate l'ordine RUN (in inglese vuol dire "gira") e battete il tasto di a capo RETURN per iniziare l'esecuzione del programma.

Notate in particolare come vengono visualizzati i contenuti di A, B e C (cioè i numeri 25, 11 e 30) e la quantità di spazi vuoti fra i tre numeri. Modificate allora l'istruzione PRINT del programma nel modo seguente:

30 PRINT A; B; C

e osservate la spaziatura tra i vari numeri quando, invece di una virgola, tra le variabili mettiamo un punto e virgola.

Per modificare i dati, sarà sufficiente sostituire i numeri della dichiarazione DATA. In alternativa a READ e DATA potete usare altresì il comando INPUT. Sostituite, ad esempio il READ e il DATA del programma precedente con la seguente istruzione:

10 INPUT A, B, C

(cancellate la linea 20 digitando 20 e battendo il tasto RETURN). Quando il programma è lasciato, il VIC visualizzerà un punto interrogativo (?) al fine di invitarvi ad immettere i dati dalla tastiera.

Per far sì che il computer visualizzi un testo o un titoletto è sufficiente inserire le parole i numeri o i segni grafici desiderati tra due doppie virgolette subito dopo l'istruzione PRINT, come nell'esempio seguente:

28 PRINT "A", "B", "C"

30 PRINT A, B, C

(Nota: le virgolette di apertura e di chiusura utilizzate dal computer sono sempre le stesse).

Un ulteriore modo per identificare i tre numeri consiste nel far visualizzare (o stampare) le tre lettere seguite dal segno = e dal numero corrispondente. Sperimentate questo tipo di istruzione cercando di variare la spaziatura a volontà (ma prima cancellate la linea 28).

30 PRINT "A =" ;A; "B = ;B; "C = ";C

L'informazione contenuta tra doppie virgolette viene data in uscita dal computer esattamente come nella istruzione 30, mentre le lettere A, B e C *non* incluse tra virgolette si riferiscono alle celle di memoria. Se A, B e C contengono rispettivamente 25, 11 e 30, sullo schermo il risultato della istruzione 30 sarà:

A = 25 B = 11 C = 30

Volendo, potete inserire degli spazi ulteriori (□ qui indica uno spazio) tra le virgolette. Ad esempio, sostituendo

"□ □ □ □ □ A □ ="

nell'istruzione 30, avremmo che la lettera A verrebbe visualizzata (o stampata) dopo cinque spazi vuoti e con uno spazio bianco prima del segno # uguale.

La funzione TAB serve a far sì che una data informazione venga stampata o visualizzata in una particolare colonna, come illustra l'esempio seguente:

30 PRINT TAB (3); "A =";A; TAB (10); "B ="; B; TAB (16); "C=";C

Il risultato sarà che A = verrà visualizzato sulle posizioni 4,5 e 6, seguito dal contenuto della cella A; B = risulterà nelle posizioni 11 e seguenti, dopo di che avremo il contenuto della cella C; infine alle posizioni 17, 18 e 19 avremo C = e dalla 20 in poi il contenuto della cella C.

La funzione TAB verrà riesaminata nei capitoli seguenti.

Variabili stringa

Per molti problemi occorre immettere, memorizzare ed ottenere in output delle informazioni variabili consistenti in un misto di lettere, numeri e caratteri speciali, inclusi gli spazi. Queste serie di simboli sono dette *stringhe.* Le stringhe possono essere memorizzate in *variabili stringa.* Ad esse occorre dare un nome consistente in una lettera dell'alfabeto (eventualmente seguita da un'altra lettera o da un numero da 0 a 9), accompagnata ancora dal segno di dollaro $ come in A$, B$, C$,... Z$.

Le informazioni costanti date fra doppie virgolette sono dette *costanti stringa.*

Le variabili stringa sono essenziali per leggere e lavorare con gli archivi di informazioni soprattutto nelle applicazioni gestionali.

Ecco alcuni esempi sull'uso delle variabili stringa.

Dopo che un programma è stato scritto e ci si è accertati della sua corretteza, esso può essere usato all'infinito con dati diversi, in diverse occasioni e da diverse persone. E' consigliabile, tuttavia, prendere nota delle date in cui il programma è stato utilizzato e da chi. Per immettere questa informazione (e poi leggerla) possono essere usate due variabili stringa.

Cambiate, ora, ed inserite le istruzioni occorrenti per ottenere in output tre numeri come segue:

```
10 INPUT A, B, C, D$, N$
26 PRINT
27 PRINT "DATA" ;D$;" "; N$
28 PRINT
```

Quando questo programma è messo a punto e gira in risposta al ? che occhieggia sullo schermo dovrete immettere tre numeri, separati da virgole (per A, B e C), seguiti dalla data e dal vostro nome. Ad esempio i dati di input per il programma superiore potrebbe essere:

? 25, 11, 30, 26/08/82, L. ROSSI

Provate a far girare questo programma con informazioni e date diverse e col vostro nome. Fate attenzione al fatto che le istruzioni delle linee 26 e 28 producono solo delle righe bianche.

Perchè un testo venga stampato nel modo voluto

E' importante progettare degli output adeguati in modo che si adattino ai formati ed agli obiettivi più diversi. Per illustrare ciò vedremo i vari sistemi per l'uso delle informazioni anagrafiche (nome ed indirizzo). Il programma della tabella 2.1 consente di immettere un titolo (Sig., Sig.ra, Sig.na, ecc.), un nome ed un indirizzo, memorizzandoli in celle di memoria identificate da variabili stringa e di ottenere la stampa d'una intestazione di lettera, un'etichetta per l'agenda o un'etichetta per la busta della lettera.

```
10 INPUT T$
20 INPUT N$
30 INPUT A$
40 INPUT B$
50 INPUT C$RECT
60 INPUT D$
61 PRINT"●":REM CANCELLA SCHERMO
70 REM INTESTAZIONE LETTERA
80 PRINT  TAB(3);A$
90 PRINT  TAB(3);B$
100 PRINT TAB(3);C$
110 PRINT TAB(3);D$
120 PRINT
130 PRINT
140 END
150 REM ETICHETTA PER AGENDA
160 PRINT
170 PRINT
190 PRINT"************"200 PRINT
210 PRINT
220 PRINT TAB(5);N$
230 PRINT
240 PRINT
250 PRINT"************"260 PRINT
270 PRINT
280 END
290 REM ETICHETTA PER BUSTA
300 PRINT
310 PRINT
320 PRINT
330 PRINT TAB(5);T$;" ";N$
340 PRINT TAB(5)A$
350 PRINT TAB(5)B$
360 PRINT TAB(5)C$
370 PRINT TAB(5)D$
380 END
```

A seguito delle sei istruzioni INPUT della tabella 2.1 il computer vi richiederà di immettere sei righe di dati. Lavorando in modo interattivo, ciascuna linea di dati viene immessa in risposta al ? che lampeggia sul video (vedi tabella 2.2).

Immettete e fate girare questo programma sul vostro VIC. Memorizzatelo su cassetta battendo SAVE "TABLE 2.1" più il tasto RETURN. Il VIC vi dirà tutto quello che dovete fare.

Le dichiarazioni REM (iniziale di remarks = note) delle linee 61, 70, 150 e 290 della tabella 2.1 vengono listate col programma al solo scopo di spiegare le attività del programma stesso. Da notare che potete inserire più di una dichiarazione separandola con un due punti (:) dopo un numero di istruzione (come mostrato nell'istruzione 61). Ciò evita di usare ulteriori numeri di istruzione di linea come si mostra in diversi esempi riportati in questo stesso volume. Il comando di "pulizia dello schermo", mostrato tra virgolette nella riga 61 è ottenuto pressando contemporaneamente i tasti SHIFT e CLR/HOME. Ciò cancellerà dallo schermo tutto ciò che avevate scritto in precedenza in modo che l'intestazione della lettera venga visualizzata in cima al campo del video. Quando il programma arriva alla istruzione END del rigo 140, l'esecuzione del programma si arresta; ma, se volete, potete far proseguire l'elaborazione digitando il comando CONT (= continua). Analogamente si può fare alla fine della parte, relativa all'etichetta dell'agenda?

```
SIG.
? L. ROSSI
? VIA DEL CORSO, 45
? 20148 MILANO
?
?
```

Tabella 2.2 Dati per il programma

L'etichetta per l'agenda conterrà solo il nome del proprietario compreso tra due file di asterischi. Tuttavia in questa fase non riuscirete a centrare il nome, cosa che vedremo nel capitolo 5.

Se avete seguito attentamente il contenuto di questo capitolo dovreste essere in grado di rispondere alle seguenti domande. Misurate i vostri progressi confrontando le vostre risposte con quanto esposto nei due capitoli precedenti.

Autotest

1. Spiegate cosa è una variabile in basic.
2. Quale comando dovete impartire per far girare un programma in basic?
3. Come è possibile cambiare un'istruzione in un programma basic?
4. Come cancellare un'istruzione?
5. Come inserire una nuova istruzione?
6. Se sul visore, durante l'esecuzione del programma compare un solo ? cosa vuol dire?
7. In quale posizione vengono evidenziati i numeri se tra le variabili di un'istruzione PRINT ci sono delle virgole?
8. In quale posizione vengono visualizzati i numeri se tra le variabili di un'istruzione PRINT abbiamo dei punti e virgola (;)?
9. Come vengono rappresentate in basic la variabili stringa?
10. Cosa sono le variabili stringa e come vengono usate?
11. A cosa serve la funzione TAB? Fate un esempio.
12. A cosa serve una dichiarazione REM?

Commodore. un leader indiscusso della microinformatica:

meglio se dalla CO.R.EL.

HARDWARE

CO.R.EL. FRIULI e CO.R.EL. ITALIANA: due organizzazioni tecnico - commerciali di collaudata esperienza ed elevata professionalità, per una consulenza qualificata ed una impeccabile assistenza post - vendita.
Fra i migliori specialisti di informatica hanno selezionato questi partners:

C.B.G. COMPUTERS - Treviso - 0422/60261
DI SABATINO - Caerano S.M. - 0423/85215
MICROCOMP - Trieste - 040/775131
SIEL - Pordenone - 0434/24715
ZEROUNO COMPUTERS
Montebelluna - 0423/23440
VOLPATO - Castello di G. - 0423/43391
VECOMP - Verona - 045/594344
NEW COMPUTERS - Verona - 045/595477
O.E.I. - Padova - 049/23701
VIP - Padova - 049/652749
IBIS - Padova - 049/38376
CORMOE - Venezia - 0421/53831
ELETTRONICA BISELLO - Vicenza - 0444/512985
LA TECNICA UFFICIO - Bassano del G. - 0424/22413

Agenzia CO.R.EL - Trieste
040/762272
Agenzia CO.R.EL - Pordenone
0434/33158

CO.R.EL FRIULI
ogni anno produce per il mercato europeo migliaia di interfacce per computers.

SOFTWARE

Dalla CO.R.EL FRIULI anche programmi applicativi, realizzati da professionisti per i professionisti dell'ingegneria civile, termotecnica, della gestione di imprese di costruzioni.
Programmi di livello superiore, apprezzati e diffusi in tutta Italia, e distribuiti da questi specialisti:

STUDIO SYSTEM - Perugia - 075/754964
GIANNONE - Modica - 0932/904705
SAIU ELETTRONICA - Frosinone - 0775/83093
COPY DRY - Napoli - 081/322765
STAR DATA - Perugia - 075/33312
INFORMATICA POINT - Casarano - 0833/331619
INFORMATICA & COMPUTERS
Corato - 080/824133
R.A.M.E.D. - Siracusa - 0931/22700
MASELLI PER UFF. - Foggia - 0881/76111
GRAAL SYSTEM - Salerno - 089/321781
INFORMATICA SISTEMI - Melfi - 0972/64412

E dalla CO.R.EL. ITALIANA procedure interattive gestionali di facile apprendimento ed alta professionalità, per l'industria ed il commercio.

CO.R.EL - Via Mercatovecchio, 28 - tel. 0432/291466 - UDINE
Distributrice ufficiale dei prodotti COMMODORE per il Friuli V.G. e per il Veneto

Invasione spaziale

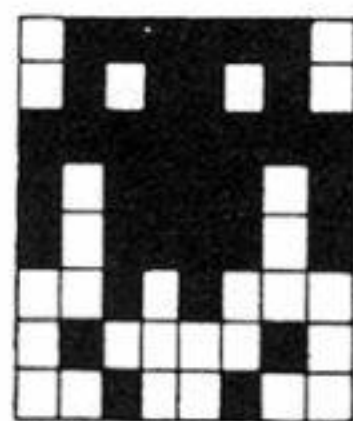
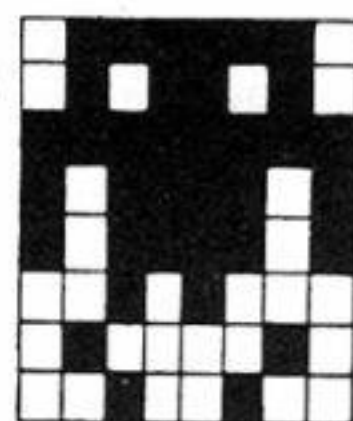
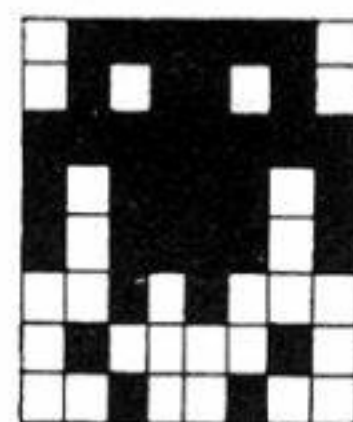
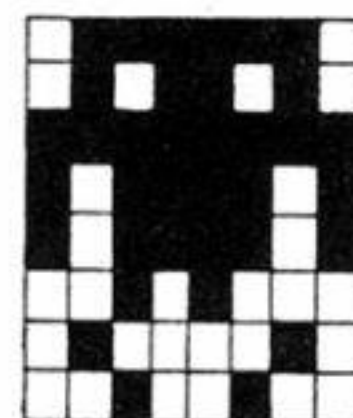
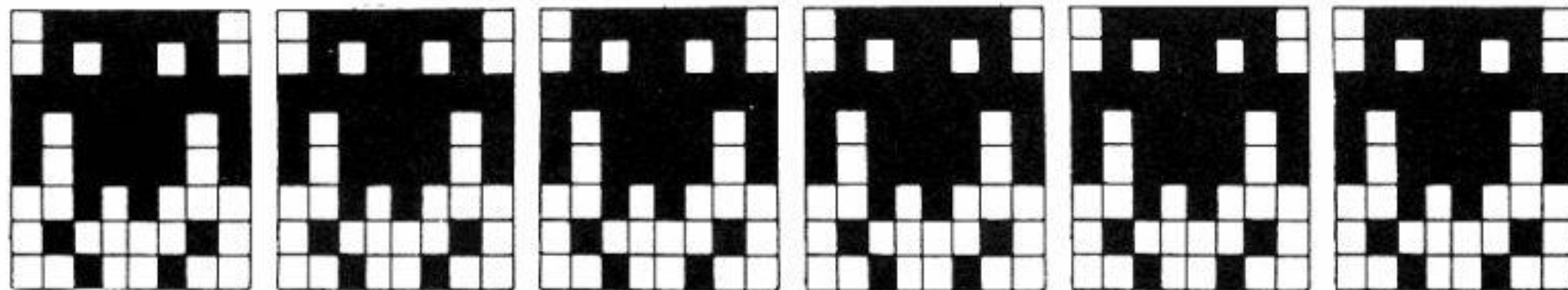
QUESTA piccola versione del noto gioco "Invasione spaziale" ha il pregio di evidenziare alcune interessanti caratteristiche grafiche del VIC. Come gioco non è eccezionale: rassomiglia più ad un tiro al bersaglio che ad una invasione di extraterrestri. Gli invasori infatti non possono nè scomparire nè difendersi ed il giocatore deve evitare soltanto di esaurire le sue riserve di pallottole. Tali riserve sono visualizzate in rosso in testa lo schermo e man mano che voi sparate diminuiscono. Se continuate a tirare anche dopo il loro esaurimento non succede nulla: il gioco, però, visualizzerà il vostro punteggio e vi darà l'opportunità di riprendere la partita.

I controlli sono buoni:

P = muove a sinistra;

@ = è il grilletto col quale fate fuoco con la vostra pistola a laser;

* = muove a sinistra.

```
5 ** PRIMI BIT **
10 POKE650,128
15 DIMA%(4):SC=0
20 TI$="000000"
25 GOSUB12000
30 ** BLOCCO PRINCIPALE **
40 GOSUB5000
50 GOSUB4000
60 GOSUB6000
70 GOSUB7000
80 GOSUB8000
82 REM ** CERCA GLI ALIENI **
85 IFNA=0THEN11000
90 GOTO70
3998 END
3999 REM ***SUBROUTINES ***
4000 REM INIZIALIZZA
4010 FU=20:FU$="[illegible]●●●●●●●●●●●●●●●●●●●●●●●"
4020 DIMI,J:SP$="[illegible]                         "
4025 DI=1:CO=0:NA=20
4030 POKE36878,8
4040 FORI=1TO4:A%(I)=3:NEXT
4050 LM$="[illegible]"
4060 RETURN
5000 REM ** GRAFICA**
5005 POKE56,28:POKE52,28
5010 FORI=7168TO7168+55
5020 READD:POKEI,D:NEXT
5030 DATA126,90,255,189,189,36,66,36
5040 DATA36,126,219,219,219,126,36,195
5042 DATA24,24,24,24,24,24,90,219
5044 DATA7,15,31,31,31,63,127,255
```

```
5046 DATA219,219,219,219,255,255,255,255
5048 DATA224,240,248,248,248,252,254,255
5049 DATA0,8,8,8,8,8,8,0
5050 DATA0,0,255,255,255,255,0,0
5055 FORI=7424TO7431:POKEI,0:NEXT
5060 POKE36879,238
5070 PRINT"■■";:POKE36869,255
5080 RETURN
5999 REM ** RISETTA SCHERMO **
6000 AM$="■■■■■"
6005 PRINT"■■■●●●●●●●●●●●●●●●●●●●●●●●   ";
6007 PRINT"■■―――――――――――――――――――――";
6010 PRINT"■";AM$:PRINT
6020 PRINT"■    @  @  @  @  @"
6030 PRINT"■■     A  A  A  A  A"
6040 PRINT"■■     @  @  @  @  @"
6050 PRINT"■■     A  A  A  A  A"
6080 PRINT"■■■■■■■■■■■■■■■■■■■■■■■  B "
6090 PRINT" CDE■"
6100 RETURN
6999 REM **MUOVI GLI ALIENI**
7000 PRINTAM$;
7010 X=INT(4*RND(1)+1):FORI=1TOX:PRINT"■■";
     :NEXT
7020 IFDI=1ANDA%(X)<5THEN7100
7030 IFDI=-1ANDA%(X)>1THEN7200
7040 DI=-DI:GOTO7300
7100 PRINTCHR$(148):A%(X)=A%(X)+1:GOTO7300
7200 PRINT"■";CHR$(20):A%(X)=A%(X)-1
7300 REM
7500 RETURN
7999 REM **MUOVI GIOCATORE**
8000 GETYM$:IFYM$<>"P"ANDYM$<>"@"ANDYM$<>"*"
     THENRETURN
8005 IFYM$="@"ANDGH<>5THEN8080
8010 IFYM$="P"ANDCO>0THENCO=CO-1:GOTO8040
8020 IFYM$="*"ANDCO<16THENCO=CO+1:GOTO8040
8030 RETURN
8040 FU=FU-.075
8050 PRINTLM$;TAB(CO)"  B  ■■■■■■ CDE ";
8070 RETURN
8080 BP=8120+CO+2:BC=BP+30720:FU=FU-.35
8090 GOSUB7000
8095 IFPEEK(BP)=0ORPEEK(BP)=1THEN8200
8100 POKEBP,6:POKEBC,0
8120 GH=5:GOSUB9000:GOSUB8000:GH=0:POKEBP,32
8130 BP=BP-44:BC=BC-44:IFBP<7790THENRETURN
8140 GOTO8090
8200 POKEBP,8:POKEBC,10:POKE36877,190:FORI
     =15TO0STEP-.2
8205 POKE36878,I+144:NEXT:POKE36877,0
```

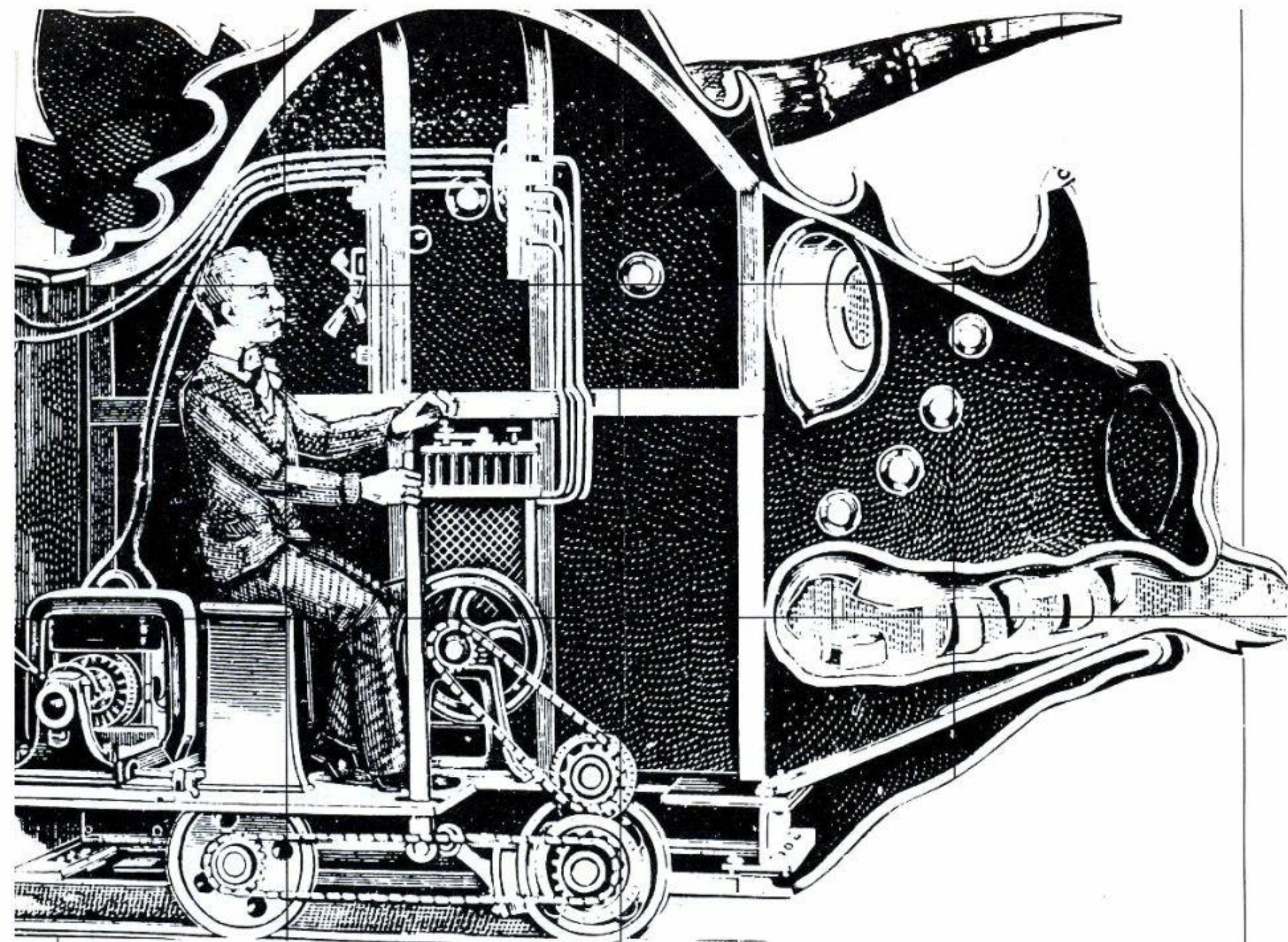
```
8210 POKE36878,8:POKEBP,32:NA=NA-1:SC=SC+10
8220 PRINT"■■■■PUNTI:"SC:RETURN
9000 IFFU<=0THEN10000
9010 PRINTSP$;LEFT$(FU$,FU+4)
9020 RETURN
9999 REM **ERRORE**
10000 PRINTSP$:FORI=1TO1000:NEXT:PRINT"■■";
10005 POKE36879,170:POKE36878,15:POKE36869,
      240
10010 PRINT"■■■■ IL CARBURANTE E' FINITO  "
10015 FORI=1TO1000:NEXT
10020 POKE36874,215:FORI=1TO500:NEXT
10030 POKE36874,207:FORI=1TO375:NEXT
10040 POKE36874,219:FORI=1TO125:NEXT
10050 POKE36874,215:FORI=1TO500:NEXT
10060 POKE36874,207:FORI=1TO500:NEXT
10070 POKE36874,0
10072 PRINT"■■HAI FATTO""SC"PUNTI"
10080 PRINT"■ANOTHER GO ? (Y/N)":POKE198,0
10090 GETA$:IFA$<>"Y"ANDA$<>"N"THEN10090
10100 IFA$="Y"THENRUN
10110 POKE650,32:POKE36879,27:PRINT"■■";:END
10160 GOTO10080
10999 REM **RIUSCITA**
11000 FORI=1TO1000:NEXT
11005 POKE36878,15
11010 PRINT"■":FORI=128TO254:POKE36876,I
11015 POKE36879,I:NEXT:POKE36879,93:
      POKE36869,240
11020 PRINT "  BENISSIMO "
11030 PRINT"■■■  HAI COLPITO TUTTI "
11040 PRINT"■      ALIENI IN ■"
11050 PRINTMID$(TI$,3,2)" MINUTI"
      :PRINTRIGHT$(TI$,2)" SECONDI■"
11060 GOTO10080
12000 POKE36879,107
12010 PRINT"■■"TAB(5)"■■――――――――"
12020 PRINTTAB(5)"■■■■ ■ ■ ■ ■ ■"
12030 PRINTTAB(5)"■■■ ■ ■ ■ ■■■"
12040 PRINTTAB(5)"■■■ ■ ■ ■ ■■■"
12050 PRINTTAB(5)"■■ ■ ■ ■ ■■■"
12060 PRINTTAB(5)"■■■ ■ ■ ■ ■■■■"
12070 PRINTTAB(5)"■■■ ■ ■ ■ ■ ■■■"
12080 PRINTTAB(5)"■■――――――■■■"
12090 PRINTTAB(4)"**INVASIONE SPAZIALE**"
12100 PRINT"■   ■BY ANDREW CASSON"
12110 FORI=1TO2500:NEXT
13000 RETURN

READY.
```

Che c'è nella memoria del VIC?

QUESTO programmino di grande utilità specie per chi si accinge a studiare il Linguaggio Macchina (L.M.), consente la visualizzazione, in blocchi da venti, del contenuto delle locazioni di memoria ed il loro indirizzo sia in decimale, sia in esadecimale.

L'utilizzatore dispone pertanto di un valido aiuto per la verifica dei programmi in L.M. scritti, mediante statement POKE, nelle locazioni RAM a disposizione dell'utente.

Dopo aver battuto RUN appare la richiesta dell'indirizzo della locazione di partenza che deve essere esaminata. La subroutine 320 e la 470 converte il valore decimale in esadecimale e lo trasforma in stringa (460). La 170, utilizzando la 470, esamina il contenuto delle RAM in blocchi da 20; la 320 esamina la parte alta dell'indirizzo della locazione (H.A. = high address); analogamente la 360, sempre servendosi della 470, traduce in esadecimale la parte bassa dell'indirizzo stesso (L.A. = low address). La 200 unisce L.A. ed H.A. in un'unica stringa e la 210 (finalmente!) la stampa sul video. Mediante la 230 e seguenti è possibile visualizzare le successive 20 locazioni (battendo +), quelle precedenti (battendo –), ripartire nuovamente (R) oppure terminare l'esecuzione del programma (E).

```
10 REM ***
20 REM ***        COMMODORE
30 REM ***      COMPUTER CLUB
40 REM ***
50 REM ***  VISUALIZZAZIONE DELLA
60 REM ***  MEMORIA DEL   V  I  C.
70 REM ***
80 REM ***        R.W.MOORE
90 REM ***  (TRAD. A.D.S.1982)
100 PRINT"■■VISUALIZZAZIONE DEL"
110 PRINT"CONTENUTO DELLA MEMO-"
120 PRINT"RIA IN BLOCCHI DA 20."
130 P=19:Q=0:PRINT"■INDIRIZZO DI PARTENZA?"
140 PRINT"(NORMALMENTE = 4096)":INPUTV
150 FORT=V+Q TO V+P
160 GOSUB320
170 W=PEEK(T):N=INT(W/16):GOSUB470
180 G$=E$
190 N=W-N*16:GOSUB470
200 H$=E$:J$=G$+H$
210 PRINT"■";T;"■";PEEK(T);TAB(11)J$;"■";
220 PRINTTAB(16);F$;"■":NEXT
230 PRINT"■";"R";"■";"IPARTI";"  ■+■";
240 PRINT"  ■-■";" OPPURE ";"■E■";"ND"
250 GETO$:IFO$=""THEN250
260 IFO$="R"THEN 100
270 IFO$="+"THEN P=P+20:Q=Q+20:GOTO150
280 IFO$="-"THEN P=P-20:Q=Q-20:GOTO150
290 IFO$="E"THEN END
300 GOTO230
310 REM *** SEZIONE SUBROUTINE.
320 P1=256*16:P2=256:P3=16
330 N1=INT(T/P1):N=N1
340 GOSUB470
350 A$=E$
360 N2=INT((T-N1*P1)/P2):N=N2
370 GOSUB470
380 B$=E$
390 N3=INT((T-N1*P1-N2*P2)/P3):N=N3
400 GOSUB470
410 C$=E$
420 N9=N1*P1+N2*P2+N3*P3
430 N4=T-N9:N=N4
440 GOSUB470
450 D$=E$
460 F$=A$+B$+C$+D$
470 IFN<10THENE$=CHR$(N+48):RETURN
480 IFN=10THENE$="A":RETURN
490 IFN=11THENE$="B":RETURN
500 IFN=12THENE$="C":RETURN
510 IFN=13THENE$="D":RETURN
520 IFN=14THENE$="E":RETURN
530 IFN=15THENE$="F":RETURN
540 RETURN
```

Caccia al rinoceronte

LA SCENA è rappresentata dalla giungla africana. Nascosti tra gli alberi vi sono dei rinoceronti il cui numero è visualizzato in alto sullo schermo. Tu sei il piccolo uomo bianco ai piedi dello schermo stesso ed il gioco consiste nell'attraversare la foresta e nel giungere sano e salvo nel riparo che viene individuato da un'H in alto. I tasti che puoi usare sono:

TASTI FUNZIONE
f1 = alto
f3 = sinistra
f5 = destra
f7 = basso

TASTI FUNZIONE + SHIFT
f2 = alto-sinistra
f4 = alto-destra
f6 = basso-sinistra
f8 = basso-destra

Quando capiti nel campo visivo di un rinoceronte, questo emette un "uh-uhh" e ti carica. Ciascun rinoceronte può muoversi di un solo quadratino alla volta ed il Vic attenderà la tua mossa prima di farlo avanzare come descritto (vedrai infatti "LA TUA MOSSA") sullo schermo. In effetti hai un piccolo vantaggio rispetto al rinoceronte dato che puoi muoverti in diagonale attraverso gli spazi tra gli alberi mentre l'animale non può farlo. Quando giungi (se giungi...) nel riparo, il Vic ti offre un'altra possibilità, ma stavolta i rinoceronti sono due. Quanti animali riuscirai ad evitare?

Esame del programma

Gli statement POKE usati ser-

vono per disegnare la giungla e per muovere il giocatore. Sul Vic lo schermo visualizza, carattere dopo carattere, secondo opportuno codice, i contenuti delle 506 (= 22 linee * 23 colonne) locazioni di memoria RAM il cui indirizzo parte (Vic da 5K) da 7680 (primo carattere in alto a sinistra) a 8186 (ultimo carattere in basso a destra nello schermo).

Nel programma presentato le locazioni dello schermo sono individuate in base a semplici calcoli sulle coordinate X-Y dello schermo stesso. L'indirizzo di partenza dell'area di schermo cambia a seconda dalla quantità di RAM presente sul proprio apparecchio. La linea 1001 tiene conto di questo fatto ed automaticamente calcola il nuovo indirizzo di schermo.

La parte principale del programma è rappresentata dalle linee 1010-1030 che utilizza una serie di subroutine svolgenti diversi compiti:
200 prepara lo schermo;
5000 tiene conto della mossa impostata;
6000 muove sia l'uomo che il rinoceronte;
7000 decide quando far apparire l'animale;
7100 produce gli effetti sonori;
8000 determina il movimento del rinoceronte.

La giungla è creata, come detto, con statement POKE che fanno apparire, in modo casuale quadratini bianchi rappresentanti gli alberi. Alcune linee di programma assicurano che il giocatore sia abbastanza distante dalla casa prima che il gioco abbia inizio. In seguito, nel caso debbano apparire più rinoceronti, la routine randomizza una coppia X-Y di valori per ciascun animale e li memorizza in un array (matrice).

Alla 1010 il programma chiede la mossa da effettuare e per evitare che il Vic tenga conto di pressioni di tasti quando queste non sono richieste, si è inserito il POKE 198,0 che, scaricando il buffer della tastiera, non tiene conto di pressioni indesiderate o accidentali. Ciò consente che venga preso in esame solo il primo tasto premuto dopo che appare il messaggio "LA TUA MOSSA". Alcuni statement IF-...THEN consentono di considerare unicamente i tasti funzione che appaiono nella tabellina prima vista, facendo saltare il programma alla linea 5000. Il valore del tasto premuto è convertito in un numero intero compreso tra 1 e 8 secondo la tabellina qui sotto riportata.

Un'opportuna routine evita che la pedina-uomo vada oltre i bordi inferiore, superiore, sinistro e destro dello schermo ed entra in funzione per X = 1, X = 22, Y = 1 ed Y = 21. La stessa routine controlla che la casella non sia già occupata. Per effettuare i diversi calcoli si è fatto ricorso alle due funzioni FNX ed FNY che così come sono scritte, possono far sorgere perplessità:
FNX(M)=(M=1 OR M=5 OR M=6)-(M)6 OR M=4)

Vediamo di chiarirne il funzionamento.

Quando il Basic incontra uno statement come T = (M=2) controlla se il contenuto delle parentesi è vero oppure no. Se M=2 è falso, allora T viene posto eguale a zero. Se invece M è effettivamente eguale a due, allora T viene posto a - 1. (Attenzione: meno uno e non semplicemente uno). Il lettore ha capito che analogo risultato si poteva ottenere ricorrendo a statement IF ...THEN.

Nel nostro caso, guardando la tabellina del valore di M, notiamo che i tre valori che spingono in alto la pedina sono 1, 5 e 6. Pertanto l'espressione M=1 OR M=5 OR M=6 sarà vera ma quella M=4 OR M>6 sarà falsa; in conseguenza di ciò FNX varrà -1-0=1.

In maniera analoga il Vic si comporta per FNY. Il vantaggio

VALORE DI M	TASTO	DIREZIONE
1	f1	Alto
2	f3	Alto-sinistra
3	f5	Sinistra
4	f7	Alto-destra
5	f2	Destra
6	f4	Basso-sinistra
7	f6	Basso
8	f8	Basso-destra

```
1000 NR=3:O=7680:CO=30720:S1=36874:S2=S1+1:POKES1+4,10:POKE36879,10:DIMR(20,2),V
(20)
1001 IFPEEK(4096)<>0THENO=4096:CO=33792
1002 DEFFNA(X)=O+22*X:DEFFNX(M)=(M=1ORM=5ORM=6)-(M>6ORM=4)
1003 DEFFNY(M)=(M=2ORM=5ORM=7)-(M=6ORM=3ORM=8):PRINT"** RINOCERONTE":GOSUB7300
1005 GOSUB2000
1010 NF=0:X=XH:Y=YH:CH=46:CL=1:POKE198,0:PRINT"*** LA TUA MOSSA *"
1015 GETG$:IFG$<"*"ORG$>"*"THEN1015
1018 PRINT"*              ":GOSUB5000:GOSUB6000:RN=0:XH=X:YH=Y:CH=94:CL=2:IFW>0THEN
1990
1020 IFNF<NRTHENGOSUB7000
1021 IFRN=0THENGOSUB8000:IFW>0THEN1970
1024 IFRN=0THEN1010
1025 FORI=1TO10:IFV(I)=0THEN1010
1030 AD=O+22*R(I,1)+R(I,2):POKEAD,94:POKEAD+CO,2:NF=NF+1:NEXTI:GOTO1010
1970 X=XH:AD=FNA(X)+YH-1:POKEAD,26:POKEAD+1,1:POKEAD+2,16:T=50:C=0
1972 FORP=0TO2:POKEAD+CO+P,C:NEXTP:C=2+(C=2):POKES1,129+3*C:T=T-1:IFT>0THEN1972
1974 POKES1,0:PRINT"** SGRETOLATO";:GOTO1996
1990 PRINT"**RIPARO   ";:GOSUB7300:IFNR<20THENNR=NR+1
1996 PRINT": ANOTHER GO?*":POKE198,0:W=0
1997 GETG$:IFG$<>"Y"ANDG$<>"N"THEN1997
1998 IFG$="Y"THEN1005
1999 POKE36879,27:PRINT"*":END
2000 PRINT"*  "CI SONO"NR"RINPCERONTI"
2010 FORI=O+22TOO+483:IFRND(1)>.75THENPOKEI+CO,5:POKEI,160
2012 NEXTI
2014 FORI=O+30TOO+118STEP22:FORJ=1TO5:POKEI+J,32:NEXTJ,I
2015 FORI=O+426TOO+470STEP22:FORJ=1TO5:POKEI+J,32:NEXTJ,I
2018 POKEO+77,8:POKEO+77+CO,1:POKEO+473,46:POKEO+473+CO,1
2060 FORC=1TONR
2070 Y=INT(RND(1)*16+3):X=INT(RND(1)*8+8):R(C,1)=X:R(C,2)=Y
2075 IFPEEK(FNA(X)+Y)<>32THEN2070
2080 NEXTC:FORC=1TO10:V(C)=0:NEXT:XH=21:YH=11:RETURN
5000 A=AD:M=ASC(G$)-132
5100 IF(X=1AND(M=1ORM=5ORM=6))OR(X=22AND(M=4ORM>6))THEN5110
5102 IF(Y=0AND(M=2ORM=5ORM=7))OR(Y=21AND(M=6ORM=3ORM=8))THEN5110
5104 AD=FNA(X+FNX(M))+Y+FNY(M)
5106 T=PEEK(AD):IFT<>32THENW=2:IFT<>8THEN5110
5108 RETURN
5110 AD=A:M=0:W=0:RETURN
6000 AD=FNA(X)+Y:POKEAD,32
6010 X=X+FNX(M):Y=Y+FNY(M)
6020 AD=FNA(X)+Y:POKEAD,CH:POKEAD+CO,CL:RETURN
7000 FORI=NF+1TONR:X=R(I,1):Y=R(I,2):IFV(I)=1THEN7060
7010 N=ABS(X-XH):IFN=0THENN=ABS(Y-YH)
7030 D=22*(XH<X)-22*(X<XH)+(YH<Y)-(Y<YH)
7040 FORJ=1TON:AD=FNA(X)+Y+J*D:IFPEEK(AD)=160THEN7060
```

```
7050 NEXTJ:IFPEEK(AD)=46THENRN=1:V(I)=1:GOSUB7100
7060 NEXTI:I=1:IFRN=0THENRETURN
7070 IFI=NR+1THENRETURN
7080 IFV(I)=0ORI=1ORV(I-1)=1THENI=I+1:GOTO7070
7090 V(I-1)=1:V(I)=0:SX=R(I-1,1):R(I-1,1)=R(I,1):R(I,1)=SX
7095 SY=R(I-1,2):R(I-1,2)=R(I,2):R(I,2)=SY:I=I-1:GOTO7080
7100 S=150:D=8:GOSUB7200:GOSUB7200:S=130:D=16:GOTO7200
7200 POKES2,S:POKE162,0:WAIT162,D:POKES2,0:S=0:RETURN
7300 D=3:S=231:GOSUB7200:GOSUB7200:Q=Q+1:IFQ=3THEN7310
7305 S=229:GOSUB7200:GOSUB7200:GOTO7300
7310 Q=0:S=223:GOSUB7200:GOSUB7200:S=228:GOSUB7200:GOSUB7200:S=225:GOSUB7200:GOS
UB7200
7320 S=219:D=32:GOTO7200
8000 FORI=1TONR:X=R(I,1):Y=R(I,2):IFV(I)=0THENRETURN
8020 FORM=0TO8:RM(M)=PEEK(FNA(X+FNX(M))+Y+FNY(M)):NEXTM
8030 A=3:B=1:C=6:GOSUB8800:B=4:C=8:GOSUB8800:A=2:C=7:GOSUB8800:B=1:C=5:GOSUB8800
8040 FORM=0TO8:IFRM(M)=46THENW=1:GOTO6000
8041 IFRM(M)<>32THEN8044
8042 RM(M)=ABS(X-XH+FNX(M))+ABS(Y-YH+FNY(M))
8044 NEXTM:MV=110:M=0
8046 FORJ=0TO8:IFRM(J)<=MVTHENM=J:MV=RM(J)
8048 NEXTJ:GOSUB6000:IFW>0THENRETURN
8050 R(I,1)=X:R(I,2)=Y:NEXTI:RETURN
8800 IFRM(A)=160ANDRM(B)=160THENRM(C)=160
8810 RETURN
```

di ricorrere a DEF...FN è quello di risparmiare spazio in memoria.

Dopo aver valutato la mossa impostata dal giocatore, il programma continua dalla linea 6000. La variabile CH determina il carattere da stampare memorizzando la posizione corrente dell'omino e passando il turno al rinoceronte.

Un lavoro che svolge il programma è anche quello di controllare se il giocatore è capitato nel campo visivo dell'animale. Ciò è realizzato calcolando il numero di quadratini interposti tra il giocatore ed il rinoceronte. La linea 7030 determina la direzione utilizzando i principi già visti. Se XH è minore di X, l'animale è al di sotto del giocatore e se YH è minore di Y, esso è alla sua destra. D è la variabile che tiene conto della posizione dell'animale. Se sommiamo D alla posizione corrente del rinoceronte otteniamo l'indirizzo della casella in alto a sinistra, diagonalmente rispetto alla casella corrente. Il programma controlla ciascuna casella lungo tale linea finchè non incontra quella del giocatore. Se, in tale calcolo, non incontra alcun ostacolo, vuol dire che il giocatore è nel campo visivo dell'animale; a questo punto RN è resettato e la subroutine 7100 genera l'effetto sonoro. Se non è apparso alcun nuovo animale, il programma salterà alla routine 8000 che calcola la successiva mossa per ciascun rinoceronte considerando ogni possibilità ed escludendo, ovviamente, quelle che condurrebbero contro ostacoli, oltre il bordo dello schermo eccetera. La stessa routine calcola la via più breve, tra le varie possibili, per raggiungere il giocatore.

Il programma presentato gira perfettamente sul Vic da 5k, ma eventuali espansioni di memoria non richiedono alcuna modifica per un suo uso corretto.

Mapvol

di J.G.R.

ESISTE una parte di un blocco, sui dischetti formattati con le unità tipo 2031, 2040, 3040 o 4040, che dice al DOS (Disk Operating System) quali blocchi del disco stesso siano occupati e quali invece siano liberi e quindi utilizzati.

La parte di blocco in questione si trova nella traccia 18 e possiede il numero 00, cioè proprio il primo blocco della traccia.

La parte di questo blocco interessata si chiama BAM, acronimo del termine inglese Blocks Avalable Map, ovvero mappa dei blocchi disponibili.

Ogni traccia del disco occupa 4 bytes della BAM; i 4 bytes visti in maniera binaria del DOS forniscono l'esatta situazione della singola traccia.

Il DOS accede numerosissime volte alla BAM, sia in fase di richiesta di fornitura di blocchi (allocazione) che in fase di liberazione di spazio (deallocazione).

Il programma MAPVOL visualizza sullo schermo del vostro computer una rappresentazione grafica della BAM.

Questa rappresentazione da l'esatta idea di come il DOS si gestisce i suoi files, sia essi di programmi che di dati. Proprio per questa ragione il programma MAPVOL può essere considerato quale utility di curiosità e di consultazione saltuaria.

Potrete sapere quando sia necessario eseguire una copia del dischetto (non il BACKUP, ma HEADER e COPY) in maniera tale d'avere i files sistemati in una migliore maniera. Potrete capire il perchè di ciò osservando la BAM prima e dopo una sostituzione.

```
100 GOTO 270
110 *
120 ********************************************************************
130 *                                                                  *
140 *        MM   MM     AA     FFFFP    V     V     OO     L          *
150 *        M M M M    A  A    F    F   V     V    O  O    L          *
160 *        M  M  M   A    A   F    F   V     V   O    O   L          *
170 *        M     M   AAAAAA   FFFFF     V   V    O    O   L          *
180 *        M     M   A    A   F          V V      O  O    L          *
190 *        M     M   A    A   F           V        OO     LLLLLL     *
200 *                                                                  *
210 ********************************************************************
220 *             VISUALIZZAZIONE GRAFICA DELLA MAPPA DI               *
230 *               UN DISCHETTO FORMATATO SU UN DRIVE                 *
240 *        CBM-2040  O  CBM-2031  O  CBM-3040  O  CBM-4040           *
250 *K****************************************************************H*
260 *
270 INPUT "DRIVE    0   ███████";D$: IF D$="0" OR D$="1" THEN 290
280 PRINT "▀";: GOTO 270
290 OPEN 15,8,15
300 PRINT#15,"I"D$: NU$=CHR$(166): Z4=1
310 OPEN 2,8,2,"#"
320 Y$="████████████████████████"
330 X$="██████████████████████████████████████"
340 DEF FNS(Z) = 2↑(S-INT(S/8)*8) AND (SB(INT(S/8)))
350 PRINT#15,"U1 ";2;D$;18;0
360 PRINT#15,"B-P";2;1
370 PRINT "▀";
380 Y=23: X=1: GOSUB 680
390 FOR I=0 TO 20: PRINT: PRINT "TT" RIGHT$(STR$(I)+" ",3);: NEXT I
400 GET#2,A$: GET#2,A$: GET#2,A$
410 TS=0
420 FOR T=1 TO 35
430 :GET#2,SC$: SC=ASC(RIGHT$(CHR$(0)+SC$,1))
440 :TS=TS+SC
450 :FOR SN=0 TO 2
460 ::GET#2,A$: IFA$="" THEN A$=CHR$(0)
470 ::SB(SN)=ASC(A$)
480 :NEXT SN
490 :Y=24: X=T+3: GOSUB 680
500 :PRINT LEFT$(RIGHT$(STR$(T),2),1) "██" RIGHT$(STR$(T),1); "█TTT";
510 :FOR S=0 TO 20
520 ::IF T<18 THEN 560
530 ::IF T>30 AND S=17 THEN PRINT NU$;: GOTO 600
540 ::IF T>24 AND S=18 THEN PRINT NU$;: GOTO 600
550 ::IF T>24 AND S=19 THEN PRINT NU$;: GOTO 600
560 ::IF T>17 AND S=20 THEN PRINT NU$;: Z4=Z4+1: GOTO 600
570 ::PRINT "█";
580 ::IF FNS(S)=0 THEN PRINT"+";: GOTO 600
590 ::PRINT "█+";
600 ::PRINT "███";
610 NEXT S,T
620 PRINT#15,"I"+D$
630 GET A$: IF A$="" THEN 630
640 IF A$="R" THEN RUN
650 IF A$="C" THEN PRINT "▀"
660 IF A$="D" THEN PRINT "████DIRECTORY D"D$"█": POKE 158,1: POKE 623,13
670 END
680 PRINT LEFT$(Y$,Y) LEFT$(X$,X) "█";
690 RETURN
```

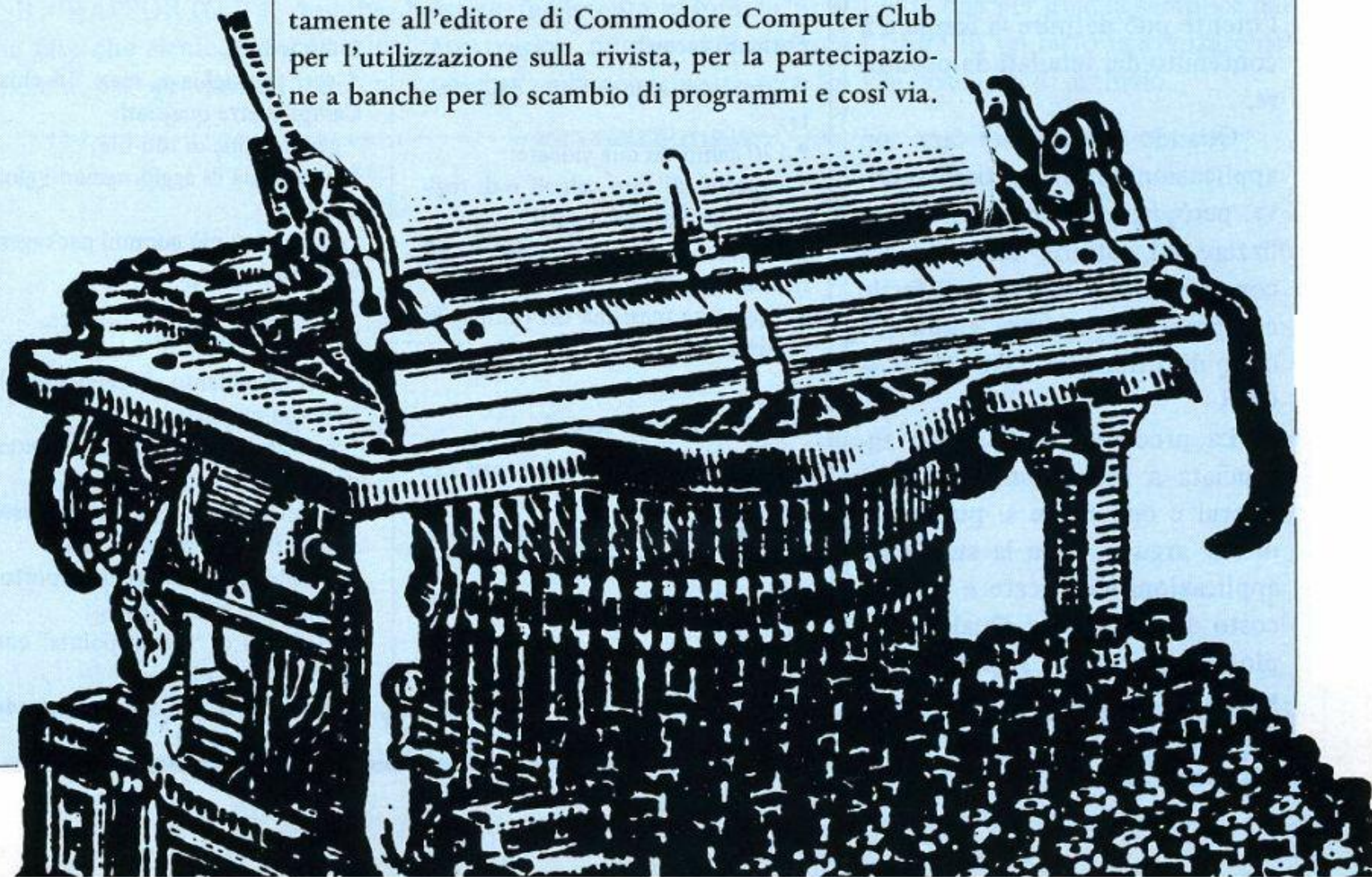
Scrivi anche tu

La collaborazione a Commodore Computer Club è aperta a tutti gli utenti di sistemi CBM che abbiano sviluppato programmi e routine originali e d'interesse generale o che desiderino fare partecipi gli altri lettori delle scoperte e delle osservazioni fatte per una migliore utilizzazione di queste macchine.

Per collaborare è sufficiente inviare i propri lavori all'attenzione della redazione, che le esaminerà tutti con uguale interesse e sollecitudine.

Gli articoli debbono pervenirci scritti a macchina o con stampante. I listati allegati debbono essere accompagnati dalla copia su cassetta o su dischetto per una rapida verifica della loro validità ed esattezza. Per compensarvi per questa spesa la redazione vi spedirà a stretto giro di posta due cassette e/o dischetti nuovi per ciascuna cassetta da voi inviata, indipendentemente dal fatto che il programma registrato venga accettato per la pubblicazione.

Ciascun articolo e/o programma pubblicato, invece, verrà compensato con l'invio d'una intera scatola di cassette e/o dischetti.

Gli autori, naturalmente, si assumono ogni responsabilità circa l'originalità dei lavori inviati i cui diritti di riproduzione si intendono ceduti illimitatamente all'editore di Commodore Computer Club per l'utilizzazione sulla rivista, per la partecipazione a banche per lo scambio di programmi e così via.

The Manager: Un database per il Pet

CON THE MANAGER anche un utente non tecnico può generare e gestire propri archivi di dati. La procedura permette all'utente di generare in modo semplice ed efficace direttamente sul video del sistema di elaborazione le maschere per acquisire i dati da tastiera; nello stesso momento, in modo a lui trasparente THE MANAGER forma la struttura di base (record) dell'archivio. Con altrettanta semplicità, rispondendo alle domande del programma l'utente può definire la forma e il contenuto dei tabulati da produrre.

Quando si ha a che fare con applicazioni molto particolari dove, però, necessita registrare, analizzare ed estrarre informazioni con modalità semplici e di facile comprensione, ebbene allora è il caso di utilizzare THE MANAGER.

La procedura può essere agganciata a programmi (in basic) esterni e ciò, come si può facilmente arguire, apre la strada ad applicazioni sofisticate e a basso costo di produzione. Quale esempio di integrazione si può citare:

• aggancio a procedure word processing;

• aggancio a procedure di contabilità per produrre situazioni clienti, fornitori, etc. più conformi alle modalità di consultazione degli utenti del sistema informativo;

• aggancio a procedure di gestione magazzino.

Se si dispone di un archivio di dati che deve essere analizzato in modo selettivo, il THE MANAGER chiede gli elementi che interessano (può essere un blocco di caratteri qualsiasi) e in base ad essi avvia il processo di selezione sull'intero archivio; l'utente è posto a confronto con le sole informazioni che soddisfano le sue richieste.

E non solo: l'utente è messo

Le caratteristiche

Formato record:

* completamente definito dall'utente;
* 120 campi su due videate;
* capacità di fare calcoli e di registrare i risultati sul record;
* capacità di visualizzare fino ad 80 risultati di calcolo;
* lunghezza massima del campo 79 caratteri.

Archiviazione del record:

* massima lunghezza del record 253 caratteri;
* numero massimo dei records 16000 (circa 720 lunghezza 253 c.);
* nessun limite al numero di archivi sul disco.

Gestione dei dati:

* calcoli aritmetici con 99 registri intermedi;
* sort multichiave, max. 16 chiavi. Campi chiave qualsiasi;
* generazione di sub-file;
* possibilità di aggiornamenti globali;
* agganci ai più comuni packages di word processing.

Selezione e rapporti:

* sort ascendente, discendente, alfabetico o numerico;
* possibili 20 criteri di selezione nello stesso rapporto;
* i dati di un record possono essere stampati su 60 righe diverse;
* possibilità di inviare il rapporto su stampante, video o disco;
* 10 livelli di "break points" con o senza sub totali;
* per ogni archivio si possono definire più rapporti.

rapporto di intervento

cliente Chianciani Mario

indirizzo Via Val di Susa n.3 cap 001450 citta' Latina

intervento n.	00003	data	23/03/82
modello	stamp.4022	matr.	71432
tipo intervento	g.	difetto	non inizializza
ore lavoro	1/2	ore sp.	====
		importo da fatturare	=========

in condizione di precisare attraverso la fase di "RAPPORTO" in quale forma e su quale unità di uscita i dati che lo interessano devono essergli presentati.

L'uscita può essere avviata su video, stampante e disco magnetico (subarchivi).

Il "RAPPORTO" è, dunque, una fase che richiede delle scelte e poichè può produrre documenti usati in ulteriori attività THE MANAGER permette di memorizzare, per impieghi ripetuti, parametri che ne hanno definito la produzione.

Il package riorganizza i dati secondo uno o più elementi. Le informazioni osbolete possono essere facilmente eliminate e, al contrario, nuove informazioni possono all'occorrenza essere aggiunte ampliando gli archivi esistenti.

Per accelerare le operazioni di ricerca è previsto l'uso del "file ad indice" costruito su qualsiasi elemento (campo) della registrazione (record). Comunque, le registrazioni possono essere visionate una per una, la semplice battitura di un tasto fa avanzare nelle posizioni di archivio.

dati tecnici

parte sostituita	oscil.8MHz	quantita'	1	costo	3.400
parte sostituita	74ls00	quantita'	1	costo	550
parte sostituita		quantita'		costo	

Se l'utente lo desidera può accedere ad una registrazione (record) fornendo al sistema il numero corrispondente alla posizione che questa registrazione (record) occupa nell'archivio (file).

Il package prevede operazioni aritmetiche sia nella fase di immissione che in quella della emissione dei dati.

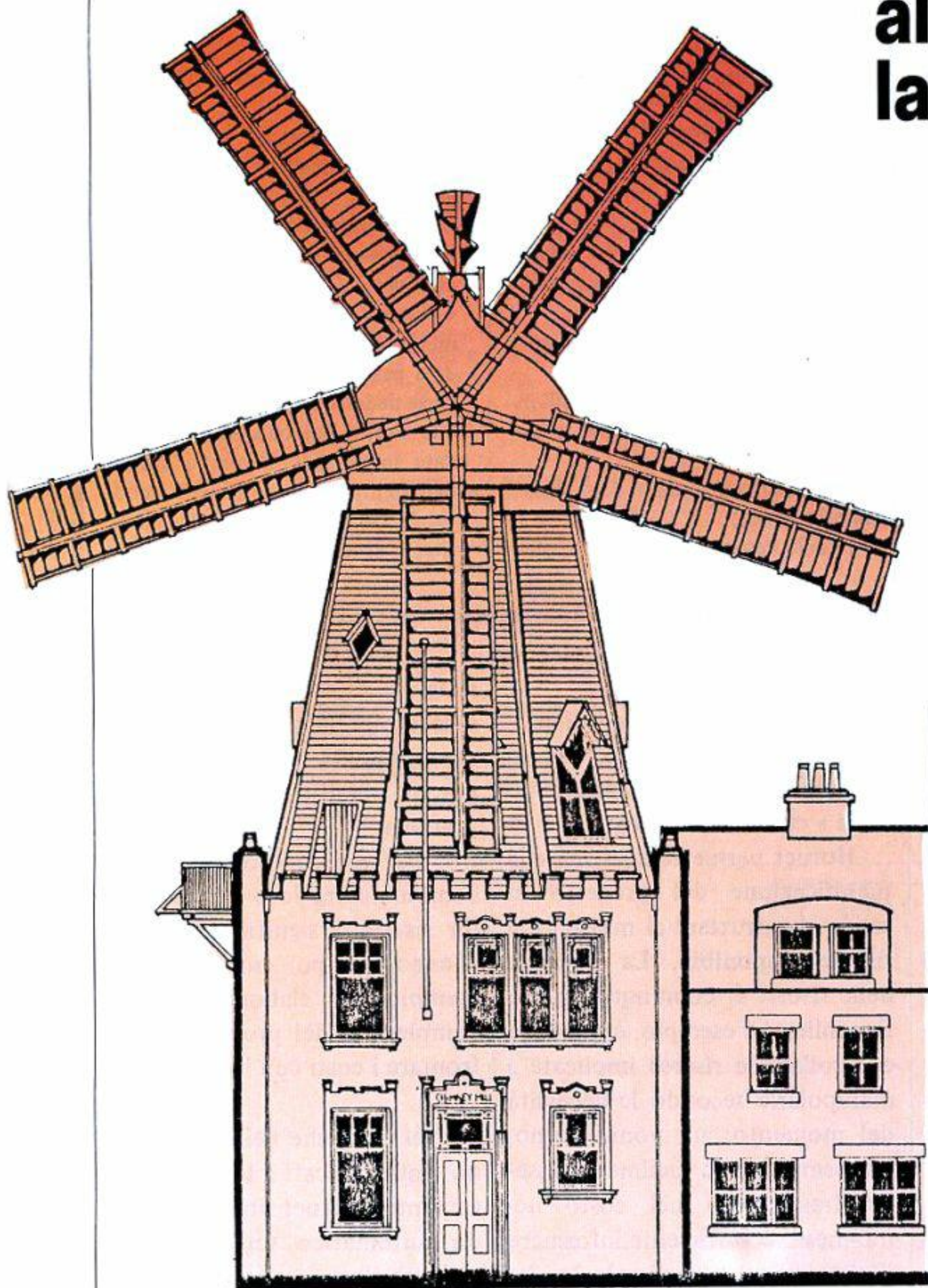
Nel primo caso i dati calcolati possono essere memorizzati nella registrazione come se fossero stati digitati direttamente dalla tastiera, ammesso che sia stato previsto lo spazio per essi (campo). Nel secondo caso i dati calcolati possono essere solo utilizzati per dare ulteriori informazioni o su video o su stampa.

In entrambi i casi THE MANAGER ha permesso di definire, nella fase di creazione della maschera video, le posizioni dove i successivi calcoli aritmetici visualizzano i risultati. Queste posizioni non sono accessibili da tastiera.

Un attento impiego delle citate possibilità consente a chiunque di generare maschere video notevolmente sofisticate.

La procedura è dotata dei seguenti programmi di utilità, tutti richiamabili da menù: formattazione nuovi dischi, backup dischi, copia di maschere video, output su video o stampante del contenuto fisico degli archivi (controllo tecnico), ampliamento archivi, copia archivi, cancellazione completa di archivi obsoleti, cancellazione da archivi di blocchi di dati obsoleti e visualizzazione directory.

```
              RAPPORTO TECNICO TRIMESTRALE

MODELLO      DIFETTO                             DATA

Vic 20       video difettoso                     07/04/82
Vic 20       rom sistema operativo               27/04/82

MODELLO


stamp.8024   interruttore alim. rotto            11/04/82

MODELLO


stamp.4022   non inizializza                     23/03/82
stamp.4022   aghi testina rotti                  09/05/82

MODELLO


pet 8032     blank video                         08/05/82
pet 8032     diminuzione byte liberi             21/04/82
pet 8032     tastiera malfunzionante             17/04/82
pet 8032     video malfunzionante                15/03/82

MODELLO


pet 4032     alimentazione difettosa             02/05/82

MODELLO


floppy 8050  fuori alineamento drives            16/04/82
floppy 8050  interfaccia IEEE difettosa          29/04/82
floppy 8050  lampeggio led continuato            18/04/82

MODELLO


floppy 4040  piastra analogica difettosa         04/05/82

MODELLO
```

Hornet: ingenere, al lavoro

Hornet è una procedura per l'analisi e la pianificazione dei progetti ed è stata scritta per lavorare su un sistema Commodore composto da 8032, 8050 e una stampante possibilmente a 132 colonne.

Con Hornet si possono analizzare sequenze complesse di attività, e determinare il massimo ritardo (se un ritardo è tollerabile) accettabile in ciascuna delle attività. Una informazione di questo tipo aiuta nella assegnazione delle priorità e quindi nella ottimizzazione delle risorse disponibili.

Le informazioni da dare alla

procedura sono poche e, al contrario, i dati che essa genera sono molti e di grandissima utilità.

All'avvio del progetto si caricano le attività in modo anche disordinato: è necessario sapere la durata stimata di ogni attività, le attività da cui essa dipende; ad esempio: questa attività non può iniziare se prima non è terminata quest'altra; questa attività può cominciare due settimane dopo l'inizio di quest'altra etc. Si aggiunga la data supposta di avvio del progetto e immediatamente il sistema produrrà la pianificazione delle date di partenza e di termine di ciascuna attività.

In ogni momento si possono inserire nuovi dati che o confermano le previsioni o introducono variazioni rese necessarie dall'avanzamento del progetto. In ogni caso Hornet ricalcolerà e ristamperà le nuove pianificazioni.

In sostanza il package crea un modello del progetto e quindi risponde ad ogni sollecitazione come lo stesso progetto; se lo svolgimento reale del progetto si è allontanato dalla pianificazione, Hornet ne tiene automaticamente conto e ne riporta le conseguenze nelle attività future. Poichè si lavora su di un modello si possono tranquillamente provare alternative diverse ed avere una visione del futuro sulla base delle pianificazioni precedenti.

Il package prevede anche la gestione delle risorse, siano esse uomini, denaro, macchine etc. Esso gestisce sino a 128 differenti risorse per progetto; su ogni attività può trattare otto diverse risorse.

Le risorse implicate nel progetto rimarranno al passo con esso indipendentemente dalle variazioni a cui sarà sottoposto.

Hornet permette di affinare la pianificazione del progetto in modo da sfruttare al massimo le risorse disponibili. La gestione delle risorse é, comunque, molto flessibile. Ad esempio, è possibile controllare le risorse implicate e manipolarle secondo le necessità del momento: un uomo-giorno, ad esempio, può facilmente essere trasformato nel costo uomo-mese: è sufficiente informare il sistema sul tipo di calcolo che si vuole effettuare sulle risorse, esso svolge l'elaborazione richiesta e presenta i risultati sotto forma di pianificazione e di istogramma.

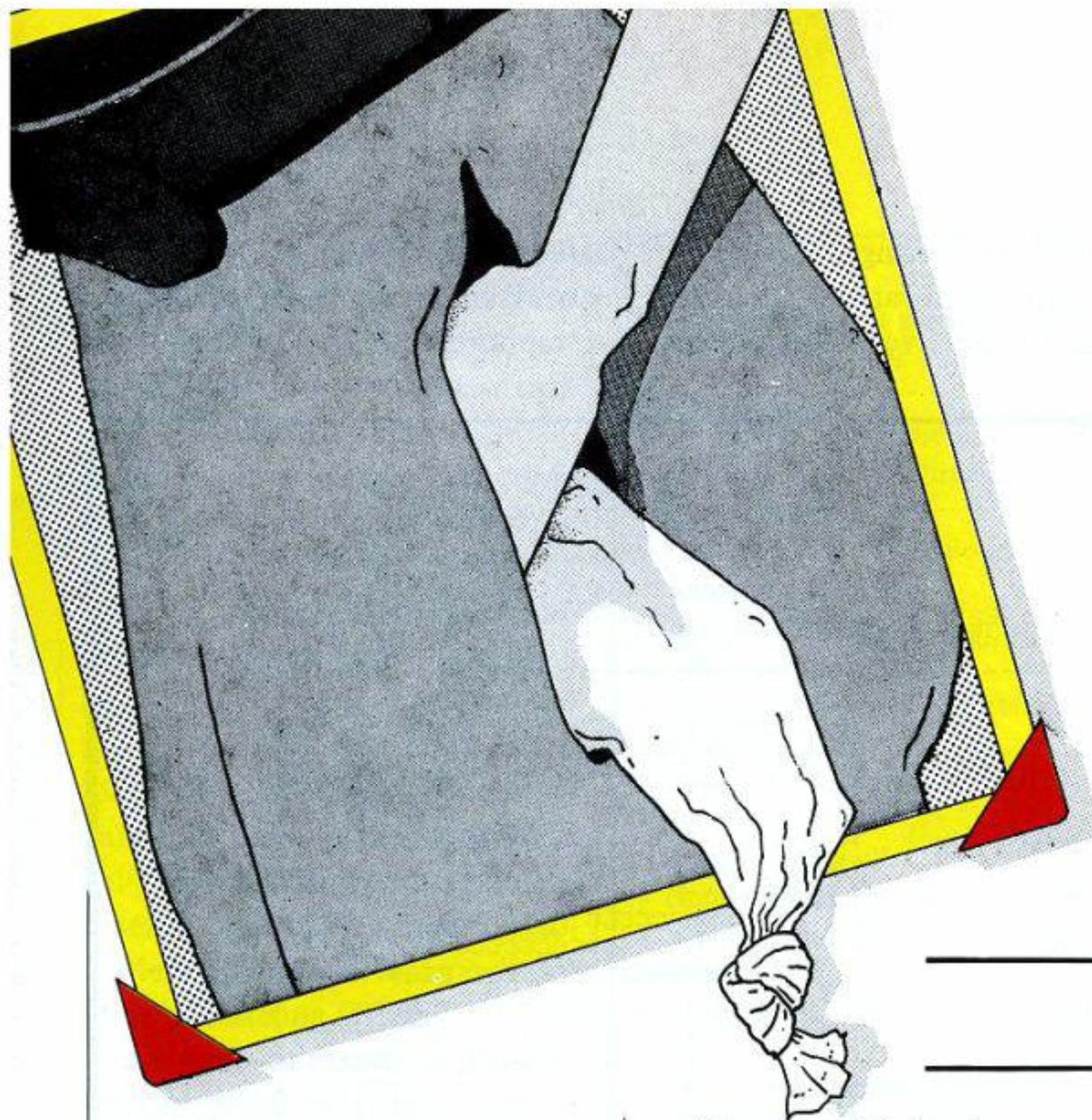
Le risorse possono essere combinate tra loro per produrre un risultato significativo e allo stesso tempo cumulativo, ad esempio per elaborare il costo complessivo del progetto o confrontare i costi con il budget.

Nel caso che nelle risorse siano stati indicati i termini di pagamento, Hornet produce in modo automatico una predizione del cash flow.

Le specifiche più significative

- 1024 attività per progetto (un floppy disk);
- possibilità di scegliere la scala dei tempi (calendario, settimane e giorni, unità arbitrarie);
- 32 periodi di interruzione (ferie) di lunghezza illimitata;
- le attività specificate non risentono delle interruzioni;
- disponibilità di 25 caratteri per descrivere la attività;
- codici di selezione definiti dall'utente;
- attività "dummy";
- cinque differenti tipi di "links";
- ritardi opzionali su "links" precedenti;
- possibilità di più avvii e completamenti;
- avvii e completamenti definibili in ogni momento per controllo avanzamento;
- data odierna per analisi della situazione odierna;
- calcolo tempi rimasti per il completamento delle attività;
- definizione degli avvii e completamenti "target";
- pianificazione dei ritardi nelle attività a motivo delle risorse;
- tempo di completamento del progetto;
- 128 tipi di risorse definibili dall'utente;
- disponibili 25 caratteri per descrivere la risorsa;
- ritardo temporale (+ / -) definibile in modo arbitrario su ogni risorsa;
- limite arbitrario delle risorse;
- si possono allocare 8 risorse per attività;
- l'utente può arbitrariamente per mezzo di 16 differenti calcoli (max) di 8 passi (max) ciascuno creare risorse dedotte da quelle di base;
- le risorse possono essere consolidate in periodi di tempo definiti dall'utente in tre differenti modi, totale per il periodo, media sul periodo, accumulazione nel periodo.

La procedura è stata programmata in modo aperto e, quindi può essere agganciata a data base o a word processing. Ciò significa che tutti i dati caricati o prodotti possono essere velocemente scaricati su archivi esterni per successiva elaborazione quale preventivi, rapporti etc.

```
-----------------------------------------------------------------------------------
HORNET   4:1    BAR CHART                                PAGE           1-OF-2
JOB NAME  : COSTRUZIONE VILLA                            CURRENT DATE   28-05-82
PEFERENCE :                                              ANALYSIS DATE  28-05-82
PROJECT COMPLETION TIME :  16 WEEKS  1  DAYS             LAST MODIFIED  28-05-82
REAL RUN DATE           :  NOT APPLICABLE       1 JUN 1982  TO  27 JUL 1982
-----------------------------------------------------------------------------------
ACTIVITY     ACTIVITY                     JUNE                     JULY           :
 NUMBER      DESCRIPTION                  :1   :8   :15  :22  :29  :6   :13  :20  :
-----------------------------------------------------------------------------------
100      SPIANAMENTO                      AAAA :    :    :    :    :    :    :    :
150      SCAVO DI FONDAZIONI              :   AA    :    :    :    :    :    :    :
160      COSTRUZIONE FONDAZIONI           :    :CCCC:    :    :    :    :    :    :
170      COSTRUZIONE FOGNE                :    :SSSS-----------------------  :    :
170.1    CONNESSIONE FOGNATURE            :    :    S------------------------------
200      MURO DA TERRA A 1 PIANO          :    :    CCCCCCCCCCCC   :    :    :    :
201      MURO DA 1 PIANO A CORNI.         :    :    :    :    : CCCCCCCCCCC  :    :
202      MURO INTERNO                     :    :    EEEEEEEEEEEEEEEEEEEEEEE----   :
250      TRAVI DI TERRA                   :    :    :    :    : SS-------------   :
251      TRAVI DEL PRIMO PIANO            :    :    :    :    :    :    :  S----  :
252      TRAVI DEL TETTO                  :    :    :    :    :    :    :   SS----:
253      SCALE                            :    :    :    :    :   S------------   :
254      PAVIMENTAZIONE                   :    :    :    :    :    :    :    :    :
255      TRAMEZZI                         :    :    :    :    :    :    :    :    :
260      TRAVICELLI TETTO                 :    :    :    :    :    :    :  CCCC   :
262      COPERTURA DEL TETTO              :    :    :    :    :    :    :    : CCC:
299      COMPLETAMENTO STRUTTURA          :    :    :    :    :    :    :    :   D:
300      IMPIANTO ELETTRICO               :    :    :    :    :    :    :    :    C
300.1    IMPIANTO ELETTRICO 2             :    :    :    :    :    :    :    :    :
300.2    CONNESSIONE RETE ELETT.          :    :    :    :    :    :    :    :    :
310      IMP. IDRAULICI INTERNI           :    :    :    :    :    :    :    :    C
310.1    CONNESSIONE RETE ACQUED.         :    :    :    :    :    :    :    :    :
320      INSTALLAZIONE GAS                :    :    :    :    :    :    :    :    S
399      IMPERMEALIZZAZIONE STRUT.        :    :    :    :    :    :    :    :    :
400      INTONACO SOFFITTO                :    :    :    :    :    :    :    :    :
401      INTONACO MURA                    :    :    :    :    :    :    :    :    :
450      INFISSI                          :    :    :    :    :    :    :    :    :
452      FINESTRE                         :    :    :    :    :    :    :    :    :
460      PREP. CUCINA                     :    :    :    :    :    :    :    :    :
462      PREP.BAGNO                       :    :    :    :    :    :    :    :    :
470      BATTISCOPA                       :    :    :    :    :    :    :    :    :
471      BATTISCOPA 1 PIANO               :    :    :    :    :    :    :    :    :
490      DECORAZIONI ESTERNE              :    :    :    :    :    :    :    :    :
600      RECINZIONE GIARDINO              :    :SS-------------------------  :    :
605      SISTEMAZIONE GIARDINO            :    :    SSSSS-----------------------  :
607      SISTEMAZIONE VIALI               :    :    :    SSS----------------------:
610      PIANTE GIARDINO                  :    :    :    :    :    :    :    :    :




------------------------------------------------------------------------SHEET 1
```

Prfile

di **J.G.R.**

QUANDO si utilizza un computer per scopi gestionali, in particolare, è inevitabile che accada il fatto di creare o comunque utilizzare dei files di dati.

Che cosa è un file di dati? Un file, termine questo prettamente inglese che tradotto in italiano significa "coda", è un insieme di informazioni scritte su supporto magnetico,oggi, su striscia di carta o su cartoncini perforati, ieri ed anche oggi, ma piuttosto raramente.

Le informazioni che compongono un file vengono scritte in alcuni modi differenti gli uni dagli altri. Il modo di scrivere una "coda" di dati costituisce essenzialmente ciò che chiamiamo "organizzazione". Il modo, l'organizzazione più utilizzata nell'ambiente dell'informatica è quella chiamata: organizzazione sequenziale.

Che cosa si intende per organizzazione dei dati in maniera sequenziale? E' presto detto.

Come si è visto un dato, un'informazione, o meglio un record (termine puramente di informatica) può essere costituito da uno o più caratteri (bytes); l'insieme di records costituisce un file. Ora se scriviamo, su nastro o su disco, un record dietro l'altro così come partono dal programma, avremo creato un file di dati ad organizzazione sequenziale.

In sintesi se il contenuto di un file è costituito da records scritti progressivamente senza una logica di chiave si dice che questo file è a struttura sequenziale.

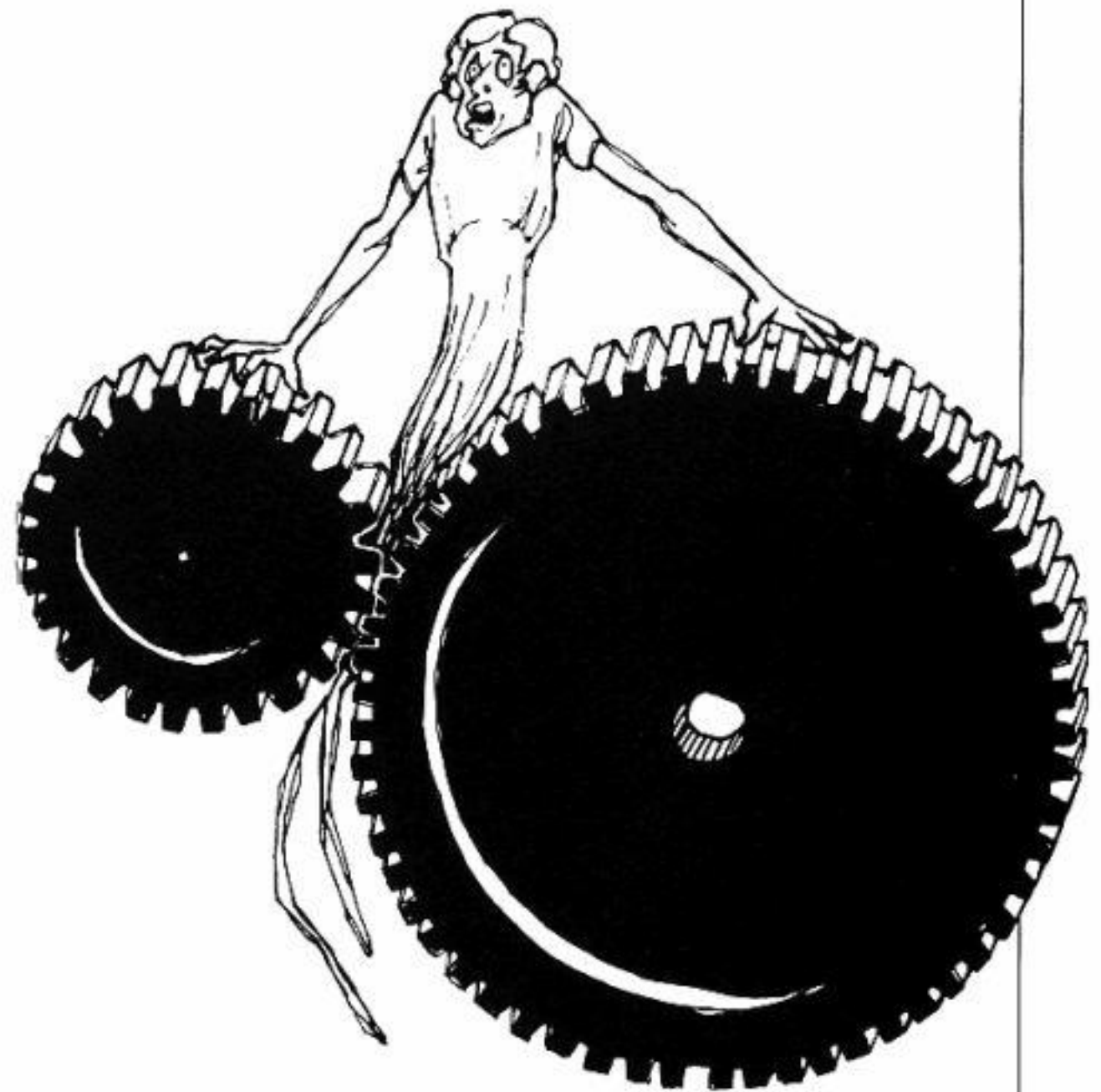
Il programma che viene proposto può essere considerato, senza alcun dubbio, una "utility", un programma cioè che può essere utilizzato in molteplici casi, non per funzionalità specifiche e ristrette, ovvero prettamente dedicato.

PRFILE può leggere qualsiasi file di dati organizzati sequenzialmente e registrati su disco e per merito di una scelta può produrre anche una copia su carta tramite la stampante oltre che solamente sullo schermo del computer.

Il funzionamento di questo programma è sufficentemente spiegato nei REMarks posti sul lato destro di ogni riga di istruzioni; è proprio per questa ragione che non mi dilungo in ulteriori delucidazioni che a questo punto risulterebbero inutili.

Copiate allora PRFILE e memorizzatelo sul dischetto di sistema dove già sicuramente avrete raggruppato tutte le vostre utilitys.

```
100 REM ***********************
105 REM *                     *
110 REM *   P R F I L E       *  PPPP   RRRR   FFFFF  I  L      EEEEE
115 REM *                     *  P   F  R   R  F      I  L      E
120 REM * LETTURA DI UN  FILE *  PPPP   RRRR   FFF    I  L      EEE
130 REM * SEQUENTIALE DI DATI *  P      R  R   F      I  L      E
140 REM *                     *  P      R   R  F      I  LLLLL  EEEEE
150 REM *K*******************H*
155 REM
160 PRINT "▯■INIZIALIZZAZIONE DISCO"
170 PRINT "■QUALE DRIVE ( 0 O 1 )"; : INPUT DN$  :REM RICHIESTA DRIVE
180 DN$=LEFT$(DN$,1): IF DN$ > "1" THEN 170       REM TEST VALIDITA' DRIVE.
190 OPEN 15,8,15                                  REM APERTURA DI SYSTEMA
200 PRINT#15,"I"DN$                               REM INIZIALIZZAZIONE DRIVE
210 GOSUB 480                                     REM TEST DISK STATUS
220 PRINT "■SU VIDEO O STAMPANTE (TV O ST)";
230 INPUT O$                                      REM INPUT RISPOSTA PER OUTPUT
240 IF O$="ST" THEN OPEN 4,5                      REM APERTURA CONDIZIONATA
250 INPUT "■NOME DEL FILE ";F$                    REM INPUT FILENAME
260 F$=DN$+":"+F$+",S,R"                          REM COSTRUZIONE STRINGA
270 REM ***********************
280 REM *                     *
290 REM *   READ      FILE    *
300 REM *                     *
310 REM ***********************
320 PRINT "■LEGGO  FILE "F$                       REM EVIDENZA NOME FILE
330 OPEN2,8,2,F$                                  REM APERTURA DEL FILE
340 GOSUB 480                                     REM TEST DISK STATUS
350 INPUT#2,A$                                    REM LETTURA RECORD
360 RS=ST
370 GOSUB 480                                     REM TEST DISK STATUS
380 PRINT A$                                      REM STAMPA RECORD
390 IF O$="ST" THEN PRINT#4,A$                    REM STAMPA REC SU STAMPANTE
400 IF RS=64 THEN 430                             REM E' FINE FILE
410 IF RS<>0 THEN 450                             REM C'E' UN ERRORE
420 GOTO 350
430 CLOSE 2:IF O$="ST" THEN CLOSE 4
440 END                                           REM FINE PROGRAMMA
450 PRINT "■ERRORE.  STATUS = "RS                 REM FILE STATUS
460 CLOSE 2                                       REM CHIUSURA FILE
470 END                                           REM FINE PROGRAMMA
480 REM ***********************
490 REM *   LETTURA CANALE    *
500 REM *      DI ERRORE      *
510 REM ***********************
520 INPUT#15,EN$,EM$,ET$,ES$                      REM LETTURA CODICI DI STATUS
530 IF EN$="00" THEN RETURN                       REM STATUS NORMALE.
540 PRINT "ERRORE DI DISCO"                       REM EVIDENZIAZIONE DI ERRORE
550 IF O$="ST" THEN PRINT#4,"■ERRORE DI DISCO"    REM STAMPA CONDIZIONATA
560 PRINT EM$,EN$,ET$ES$                          REM EVIDENZIAZIONE STATUS
570 IF O$="ST" THEN PRINT#4,EM$,EN$,ET$ES$        REM STAMPA CONDIZIONATA
580 CLOSE 2:IF O$="ST" THEN CLOSE 4               REM CHIUSURA FILE E STAMPANTE
590 END                                           REM FINE PROGRAMMA

READY.
```

Didattica: il linguaggio macchina del Pet

QUANDO si parla con i possessori di calcolatrici programmabili che intendono allargare le proprie conoscenze sui computers, ci si sente chiedere quante linee-programma e quanti dati può memorizzare un personal, evidenziando, appunto, una delle differenze notevoli dell'organizzazione dati-programmi esistenti tra i personal e le programmabili.

In effetti è più corretto dire che il microprocessore ed il sistema operativo del computer, o della calcolatrice, sono gli elementi che gestiscono i dati in un modo per cui, a livello macroscopico, è possibile parlare di dati, programmi, istruzioni, ecc.

A livello elementare, invece, un microprocessore, qualunque esso sia, tratta in modo automatico il contenuto di una successione di locazioni di memoria che di per sè possono contenere, indifferentemente, dati oppure istruzioni.

Un microprocessore, pertanto, lavora in un modo che è quello conforme alle specifiche della casa costruttrice. Chi costruisce un computer, che si basa su quel particolare processore, deve creare "attorno" ad esso il cosiddetto *firmware* rappresentato dal Sistema Operativo (O.S.) ed, eventualmente, dal linguaggio che sarà adoperato sul computer. In una macchina da calcolo si ha in definitiva:

• *L'hardware* rappresentante l'insieme di connessioni elettriche, circuiti stampati, integrati, tastiera video, periferiche le cui caratteristiche l'utente difficilmente può modificare.

• *Il firmware* è l'insieme dei circuiti integrati (I.C.) in genere a sola lettura (ROM, EPROM, PROM) contenenti l'O.S. ed il linguaggio, quando c'è. Il firmware si può definire come l'organizzazione che gestisce le funzionalità del microprocessore. Esso, differentemente dall'hardware, può essere modificato facilmente dall'utente. E' noto che in alcuni sistemi, sostituendo le ROM, è possibile far parlare il computer non più in BASIC ma, ad esempio, in PASCAL. Tipico è il caso del Pet basic 2 che può essere trasformato in Pet basic 4 mediante la semplice sostituzione di alcuni circuiti integrati (I.C.).

Altri esempi di firmware, sempre nel caso del Pet, sono rappresentati da integrati venduti da alcune ditte, da inserire negli zoccoli vuoti per I.C. presenti sul circuito stampato, per aumentare le capacità di calcolo del computer (calcolo matriciale, word processor, ecc.).

• *Il software* è quello interamente gestibile dall'utente ed è rappresentato dalle RAM, particolari I.C. in cui è possibile sia scrivere sia leggere informazioni. Quando

facciamo girare un programma scritto, ad es., in basic, in effetti è il firmware che interpreta ciò che abbiamo digitato su tastiera o introdotto da disco o nastro, ed impone al microprocessore l'elaborazione dei dati tali da soddisfare le richieste dell'utente. Ecco perchè, quando si sceglie un computer, può essere utile sapere il "peso" del linguaggio adoperato e del sistema operativo.

Grosso modo un basic da 8K è di certo meno sofisticato di uno da 16K, ma più di uno da 4K. Anche il "peso" dell'O.S. può dare un'idea della complessità, e quindi della capacità di elaborazione, di un sistema computerizzato.

Ciò premesso ci occuperemo della organizzazione e di alcune istruzioni che un microprocessore, ed esattamente il 6502, è capace di eseguire. Esso è un micro ad otto bit; ciò significa che può trattare dati od istruzioni la cui lunghezza fissa è appunto di otto bit.

Poichè il massimo valore che può essere memorizzato con otto bit è 256 (= 2 elevato alla potenza 8) sembrerebbe che possiamo disporre, appunto, di tante istruzioni.

In realtà il set di istruzioni è limitato a 151; vi sono cioè alcuni valori che per il 6502 non hanno alcun senso, se considerati come istruzioni, e se per errore gli chiediamo di elaborarli, rischiamo di ottenere la totale perdita del controllo da parte dell'O.S. per cui è necessario ripristinare le condizioni iniziali mediante un reset oppure spegnendo e riaccendendo l'apparecchio.

Prima di ogni altra cosa parleremo dell'*accumulatore*.

Esso è uno dei registri interni al 6502 di lunghezza, inutile dirlo, otto. Il micro lo utilizza come contenitore del dato che in seguito elaborerà. Per esempio, se vogliamo che un certo dato subisca una elaborazione, è necessario, ovviamente, comunicare il dato al 6502; questo, per "trattarlo," lo invierà dapprima nell'accumulatore ed in seguito lo elaborerà in base a successive istruzioni.

Oltre all'accumulatore sono presenti, sempre all'interno del 6502, altri due registri ad otto bit, X ed Y, che possono scambiare i propri contenuti con l'accumulatore. La figura 1 rappresenta lo schema dell'interno del micro relativo solo ad Acc, X ed Y; non sono cioè stati disegnati gli altri elementi che non interessano il presente articolo. Le frecce della figura indicano che vi sono istruzioni che consentono di scambiare dati tra l'accumulatore ed il registro X e viceversa, ma non tra X ed Y direttamente tra loro. Analogamente la bidirezionalità del bus indica che è possibile "caricare" Acc, X ed Y mediante opportune istruzioni col contenuto di qualsiasi locazione di memoria e viceversa.

Si ricorda che bus deriva proprio da autobus, termine davvero indovinato per indicare la parte dell'hardware che consente di caricare e trasportare dati ovunque nella memoria del computer ed anche al di fuori di esso.

Da quanto detto precedentemente risulta evidente che il microprocessore può trattare dati o istruzioni di soli otto bit di lunghezza. Prima di passare al consueto esempio applicativo, è utile ricordare che il Pet gestisce il video mediante il sistema *memory mapped:* ciascuna delle 1000 celle del video (25 righe x 40 colonne) rappresenta il contenuto di una delle 1000 celle di memoria RAM il cui indirizzo varia da 32768 a 33768 (decimale) oppure da 8000 a 83E8 se ci riferiamo invece alla numerazione esadecimale.

Per convincersene è sufficiente far girare il programmino di fig. 2.

L'istruzione basic che consente di scrivere un valore, compreso tra 0 e 255, direttamente in una locazione RAM, è appunto POKE. La stessa istruzione, scritta però in Linguaggio Macchina che utilizza la notazione esadecimale, è 8D. Tale simbolo deve essere seguito, nella memoria del calcolatore, da due byte indicanti rispettivamente la parte bassa (L.A.) e la parte alta (H.A.) dell'indirizzo in cui si desidera scrivere il contenuto dell'accumulatore (fig. 3).

Quanto asserito può lasciare perplesso il lettore: tutti gli statements del linguaggio basic sono in effetti parole inglesi, o parti di esse, di immediata interpretazione. Ci si deve però convincere che un microprocessore "parla" un linguaggio tutto suo, formato esclusivamente da 0 ed 1 e pertanto 8D, rappresentante la tra-

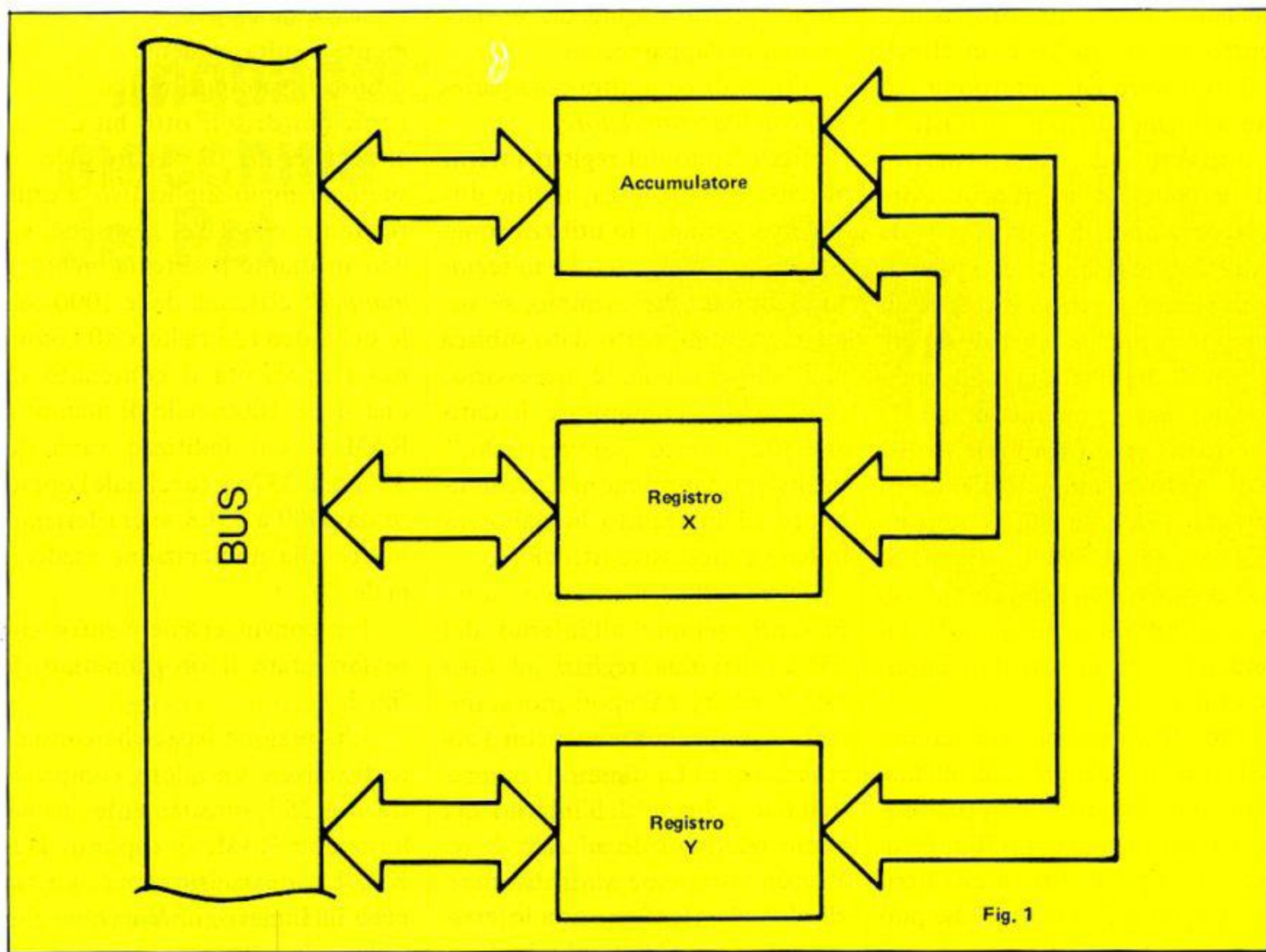



Fig. 1

duzione in *esa* che per noi non significa assolutamente nulla è per esso un preciso comando. In altre parole è questione di simboli: ognuno parla la propria lingua.

Un'altra istruzione frequentissima nei programmi in L.M. è A9. Per esempio A9 A0 viene interpretata dal 6502 nel modo seguente: carica l'accumulatore col contenuto del byte che segue immediatamente l'istruzione stessa (cioè A0). Quando il micro avrà eseguito l'istruzione (di due bytes, a differenza di 8D, istruzione a tre byte), all'interno dell'accumulatore sarà presente, nel caso specifico, il valore A0. Se carichiamo il programma di fig. 3 mediante il monitor TIM e dopo digitiamo SYS (826), verrà eseguito il programma in L.M. la cui prima istruzione è rappresentata, appunto, dal contenuto di 826 (033A), e cioè A9.

Il 6502 stabilisce, proprio in base all'ordine da noi impartito, che in 033A vi è una istruzione, non un dato, ed esegue l'elaborazione del programma partendo da tale unico presuppósto. Il micro, da parte suà, "sa" che A9 è un'istruzione a due byte, la esegue come prima detto e subito dopo esamina il byte successivo a quello contenente Ao che è 033C (in cui è 8D). In quest'ultimo caso l'istruzione è a tre byte. Dopo averla eseguita, il 6502 "incontrerà" 60 che significa "RITORNO DA SUBROUTINE". Per motivi che non stiamo qui a specificare, quando viene incontrata l'istruzione 60 senza che precedentemente, come nel nostro caso, sia stata data l'istruzione JUMP SUBROUTINE, si "ritorna" al basic.

Che cosa succede impartendo il comando SYS (826)?

1/ *A9 A0*. Nell'accumulatore viene trascritto il valore A0.

2/ *8D B5 80*. Il valore dell'accu-

mulatore (cioè Ao) viene depositato nell'indirizzo 80B5 appartenente alla mappa dello schermo: apparirà, appunto, il simbolo corrispondente ad Ao.

3/ *60.* Si ritorna al BASIC e appare il consueto READY. Notate che il 6502 stabilisce automaticamente se un'istruzione è a tre, due oppure un solo byte in quanto al suo interno è presente un decodificatore indicante il corretto procedimento da ‚guire a seconda della istruzione incontrata.

Per insistere su quanto detto digitiamo ora SYS (827) e verifichiamo che cosa accade:

• *A0 8D.* Dato che ordiniamo di partire non più da 033A ma da 033B, il 6502 stabilisce che in tale locazione vi è la prima istruzione da eseguire il cui significato è completamente diverso da A9 Ao vista nel precedente esempio. Si ricorda che A0 8D significa: carica il registro Y col valore 8D.

• *A5 80.* Carica nell'accumulatore il valore che ora è presente nella locazione 80 della pagina zero cioè 0080.

• *60.* Ritorna al BASIC.

Benchè non vi sia nulla di errato nella logica del programma così eseguito, non si verifica nulla di particolare, nè tantomeno viene stampato un simbolo sullo schermo come nel primo caso esaminato.

Digitiamo ora SYS (828).

• *8D A5 80.* Verrà trasferito, nella locazione 80A5 dello schermo, il contenuto dell'accumulatore presente in quel particolare momento dell'elaborazione. Tale contenuto può essere diverso da

```
100 INPUT A
110 FOR I = 32768 TO 33768
120 POKEI,A
130 NEXT
```

Figura 2

```
INDIRIZZI        CONTENUTO
DEC. ESA.   DELL'ACCUMULATORE

---  ----           ----
826  033A            A9
827  033B            A0
828  033C            8D
829  033D            A5
830  033E            80
831  033F            60
---  ----           ----
```

Figura 3

```
20 84 BD :RICERCA ERRORI DI SINTASSI O ALTRI
20 2D C9 :CONVERSIONE VIRGOLA MOBILE →INTERO
20 22 D7 :ROUTINE CALCOLO PARTE ALTA NUMERO
98       :TRASFERISCI Y NELL'ACCUMULATORE
20 22 D7 :ROUTINE CALCOLO PARTE BASSA NUMERO
60       :RITORNA AL BASIC
```

Figura 4

```
C*
     PC  IRQ  SR AC XR YR SP
.;  B780 E455 2C 34 3A 9D F8
.
.   033A 20 84 BD 20 2D C9 20 22
.:  0342 D7 98 20 22 D7 60 FF 00
```

Figura 4/Bis

```
100 REM ***        PROGRAMMA MISTO
110 REM ***     (LING.MACC. E BASIC)
120 REM ***         PER CONVERTIRE
130 REM ***      UN NUMERO DECIMALE
140 REM ***        IN ESADECIMALE.
150 REM ***
160 FORI=826TO839:READA:POKEI,A:NEXT
170 INPUT"NUMERO DECIMALE=";A
180 SYS(826)A
190 PRINT:GOTO170
200 DATA 32,132,189,32,45,201,32
210 DATA 34,215,152,32,34,215,96
```

Figura 5

quello da noi impostato prima, ed esattamente è quello che il Pet si ritrova dopo aver eseguito la routine di interpretazione del comando SYS (828).

• *60*. Si ritorna al basic come prima.

Anche digitando SYS (829) non avviene nulla di particolare. Battendo invece SYS (830), il 6502 cerca di eseguire l'istruzione allocata in 033E, ma poichè 80, a differenza delle precedenti istruzioni, non esiste nel set delle 151 istruzioni del microprocessore, il 6502, molto probabilmente, perde il controllo e non riusciamo più a ripristinare il sistema se non spegnendo e riaccendendo l'apparecchio.

Quest'ultimo caso, come del resto i precedenti, dimostra che è di fondamentale importanza indicare con precisione l'indirizzo di partenza del programma L.M.

Sbagliando indirizzo si ha, nel più fortunato, e purtroppo raro, dei casi, l'esecuzione di un programma diverso dalle nostre aspettative e, più spesso, la "distruzione" del sistema.

Utilizzeremo ora le poche nozioni apprese per scrivere un brevissimo programmino in L.M. che utilizza alcune lunghe e complesse routine residenti nel Pet 4032. Abbiamo prima visto che vi è la possibilità di richiamare una subroutine e la logica dell'istruzione è perfettamente analoga al GOSUB del basic. In L.M. l'istruzione GOSUB è 20 XX YY, istruzione a tre bytes in cui XX rappresenta l'L.A. ed YY, invece, l'H.A. E' ovvio che se in XXYY inizia una subroutine, essa deve terminare con un RETURN che in L.M. è rappresentato dall'istruzione implicita (che cioè non ha bisogno di specificare altri parametri per essere interpretata correttamente dal 6502) ad un solo byte il cui codice è 60.

Poichè nel Pet vi sono numerosissime sobroutine in L.M. che possono essere richiamate dall'utente per scopi particolari (a patto, ovviamente, che se ne conosca l'indirizzo esatto di partenza ed il funzionamento della stessa subroutine), descriverò il semplice programma di figura 4 utilizzante tre subroutines che, eseguite con opportuno ordine, consentono di convertire un numero decimale in esadecimale.

E' ovvio che non tenterò nemmeno di descrivere il funzionamento delle subroutine richiamate, tantopiù che ognuna di esse richiama, a sua volta, numerose altre sobroutine. Al lettore, per ora, basterà sapere che al termine di ogni subroutine richiamata vi è ovviamente l'istruzione 60 che consente di ritornare al programma L.M. principale allocato, per comodità, da 826 a 839 (fig. 5).

Nel programma di figura 4 notiamo la presenza di una istruzione che precedentemente non è stata descritta. Essa (98) consente di trasferire il valore presente in Y nell'accumulatore prima di eseguire per una seconda volta la subroutine 20 22 D7.

0 33A	20 8B CC
0 33D	20 D2 D6
0 340	20 75 E7
0 343	98
0 344	20 75 E7
0 347	60

Fig. 6

Il programma L.M. è richiamabile, mediante SYS (826) in molti modi, uno dei quali è descritto nella figura 5.

La subroutine 20 84 BD esamina che cosa è presente subito dopo SYS (826); deve, infatti, comparire un numero compreso tra 0 e 65353 oppure il nome di una variabile predefinita. In caso contrario appariranno messaggi di errore tipici del basic.

La subroutine 20 2D C9 converte il valore impostato da virgola mobile in intero. Di tale valore intero che, come il lettore sa, può occupare al massimo due bytes dato che l'intervallo trattabile è 0-65535, viene dapprima elaborata la parte alta e trascritta sul video, subito dopo, mediante la stessa subroutine, viene elaborata la parte bassa che, per motivi che non stiamo qui a dire, era stata depositata in Y prima di elaborare la precedente parte alta.

E' doveroso ricordare che la routine presentata è stata provata su di un Pet 4032 (basic 4). Chi possiede il 3032 o comunque un Pet della serie 2 deve digitare il programma di figura 6. Chi possiede il Pet vecchio ROM non può utilizzare il programma presentato perchè le subroutines sono allocate nella zona MLM (machine language monitor) di cui è purtroppo sfornito il Pet old ROM.

COMMODORE 8250: UNITA' DOPPIA PER DISCHETTI 5' 1/4

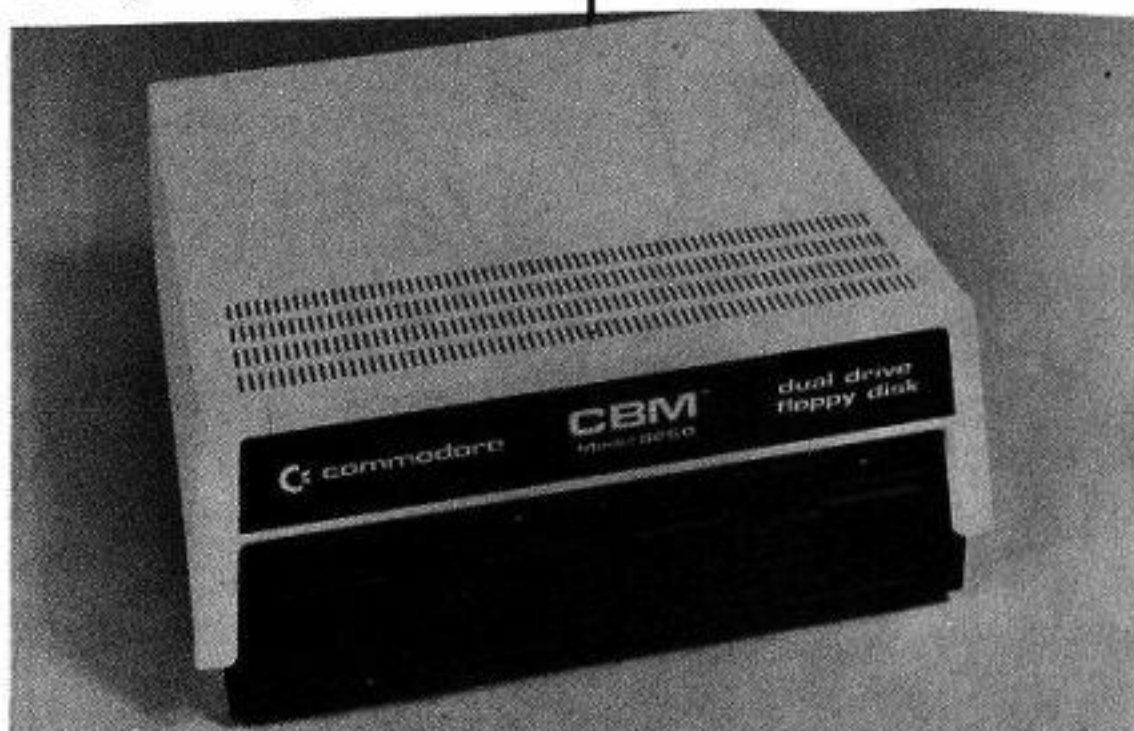
Benchè l'aspetto interno ed esterno sia molto simile al notissimo doppio drive 8050, l'8250 offre il doppio della capacità di memoria del modello minore grazie al sistema doppia faccia - doppia densità su dischetti da 5 pollici 1/4.

La superficie dei dischetti dell'8250 è organizzata in modo tale che le tracce da 0 a 77 sono allocate sulle consuete posizioni mentre quelle numerate da 78 a 154 sono situate sulla seconda superficie. La possibilità di passare dal primo al secondo blocco è completamente trasparente per l'utente finale.

Il formato del record relativo è stato espanso a 1 Mb grazie all'uso dei settori super side mentre la BAM è espansa e lo spazio destinato alla directory è riservato.

L'interscambio diretto tra i file dei record relativi generali con l'8250 e con l'8050 non è possibile, tuttavia, è disponibile un programma che consente tale operazione.

L'unità contiene all'interno un'interfaccia IEEE 488 e, come nei precedenti modelli Commodore, è intelligente e non richiede, per il corretto funzionamento, spazi di memoria nel calcolatore principale.

Il prezzo è di lire 3.450mila.

Commodore 9060 / 9090 unità a disco rigido

I nuovi drive possiedono la capacità di 5 Mb (modello 9060) oppure di 7,5 Mb (9090).

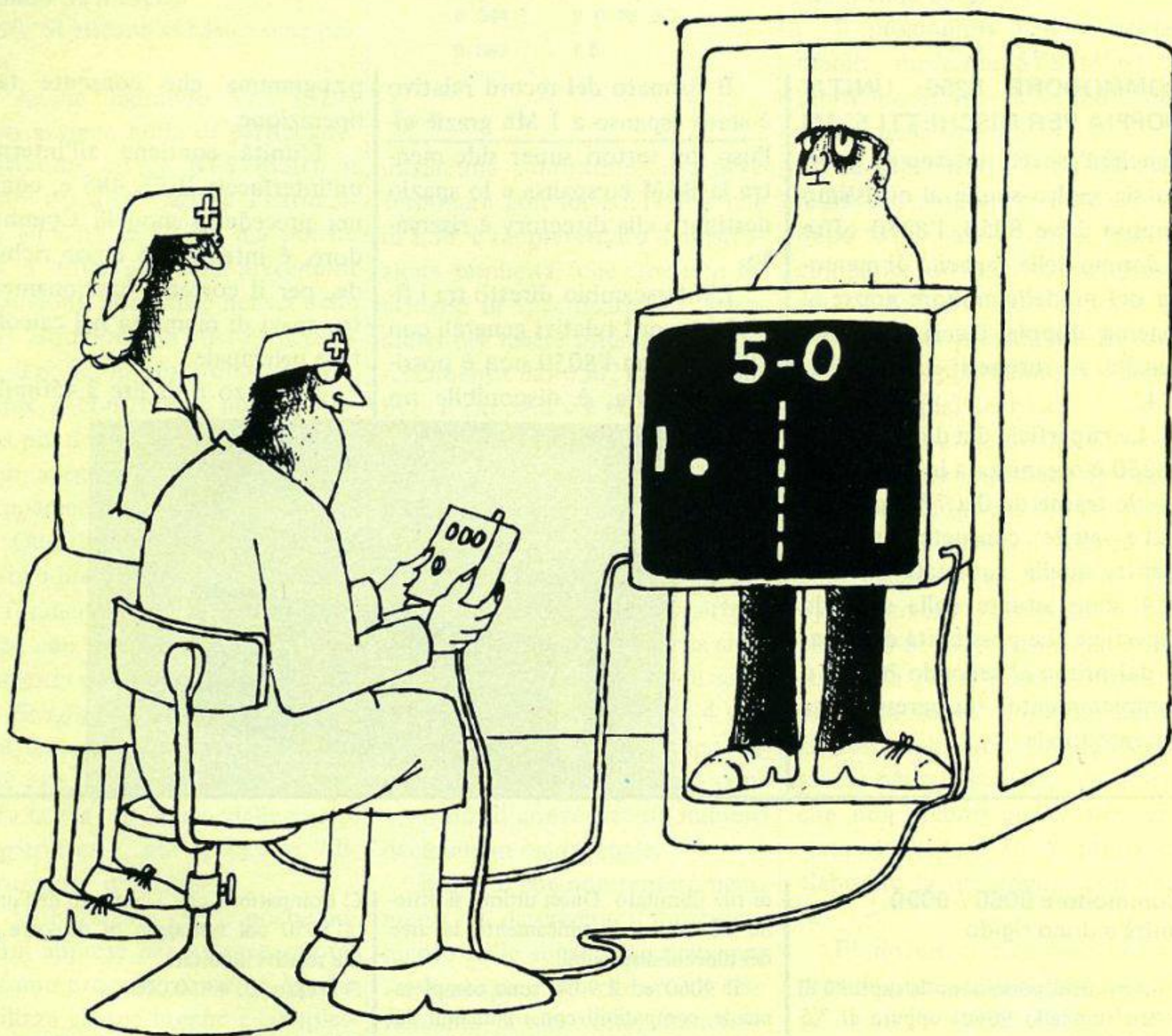
Entrambi i modelli dispongono all'interno di interfacce IEEE 488 ed usano la meccanica Tandcom per dischi rigidi. Le operazioni svolte dalle due unità sono completamente trasparenti per chiunque abbia acquisito una certa familiarità con i drive per mini-floppy 8050 e 4040 col vantaggio di una maggiore capacità e di un formato di file illimitato. Quest'ultimo si ottiene allocando dinamicamente le aree dei blocchi disponibili.

Il 9060 ed il 9090 sono completamente compatibili con i comandi dei noti standard Basic 2.0 e 4.0 ed è possibile utilizzare tutto il software precedentemente sviluppato dall'utente.

• Caratteristiche del modello 9060

□ Capacità non formattata: 6.38 Mb
□ Capacità formattata: 5.01 Mb
□ Numero dei dischi: 2
□ Numero delle testine: 4
□ Compatibile con i comandi dell'unità 8050 col vantaggio di disporre di file relativi illimitati.
□ Prezzo: L. 4.950.000

• Caratteristiche del modello 9090

□ Capacità non formattata: 9.57 Mb
□ Capacità formattata: 7.52 Mb
□ Numero dei dischi: 3
□ Numero delle testine: 6
□ Compatibile con i comandi dell'unità 8050 col vantaggio di disporre di file relativi illimitati.
□ Prezzo: L. 6.100.000.

VIC-20 computer.
Un regalo da sballo, papà!

Se lo merita un regalo, finiti gli studi, no? E allora scegli qualcosa di intelligente, che gli serva e serva a te: Vic 20, l'home computer Commodore.

Con Vic 20, 20 colori, note musicali, espandibilità da 5 a 32 k byte, il tuo ragazzo prende confidenza con la tecnologia dei computer, si diverte a programmare in Basic, a risolvere problemi didattici, scientifici e di word processing, a memorizzare dati, a giocare coi mille games possibili.
Il costo del Vic 20? Sbalorditivo, meno di 600.000 lire. E allora, via per un regalo intelligente!